



**FAKULTA  
ELEKTROTECHNICKÁ**  
ZÁPADOČESKÉ  
UNIVERZITY  
V PLZNI

VYBRANÉ PROBLÉMY Z ŘÍZENÍ  
VÍCEHLADINOVÝCH MĚNIČŮ A VÝPOČETNĚ  
EXTRÉMNĚ NÁROČNÝCH POKROČILÝCH  
ALGORITMŮ REGULACE ELEKTRICKÝCH  
POHONŮ IMPLEMENTOVANÝCH  
V HRADLOVÝCH POLÍCH

DIZERTAČNÍ PRÁCE

2014

ING. TOMÁŠ KOŠAN

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ

VYBRANÉ PROBLÉMY Z ŘÍZENÍ  
VÍCEHLADINOVÝCH MĚNIČŮ A VÝPOČETNĚ  
EXTRÉMNĚ NÁROČNÝCH POKROČILÝCH  
ALGORITMŮ REGULACE ELEKTRICKÝCH  
POHONŮ IMPLEMENTOVANÝCH  
V HRADLOVÝCH POLÍCH

DIZERTAČNÍ PRÁCE

k získání akademického titulu doktor

v oboru

Elektronika

Autor:

Ing. Tomáš Košan

Školitel:

prof. Ing. Zdeněk Peroutka, Ph.D.

Datum státní doktorské zkoušky:

20. května 2011

Datum odevzdání práce:

10. listopadu 2014

## Prohlášení autora

Předkládám tímto k posouzení a obhajobě dizertační práci vypracovanou na závěr doktorského studia na Fakultě elektrotechnické Západočeské univerzity v Plzni. Prohlašuji, že jsem dizertační práci na téma **Vybrané problémy z řízení vícehladinových měničů a výpočetně extrémně náročných pokročilých algoritmů regulace elektrických pohonů implementovaných v hradlových polích** vypracoval samostatně za použití odborné literatury a pramenů uvedených v seznamu na konci této práce.

V Plzni, 10. listopadu 2014

.....  
Ing. Tomáš Košan

## Prohlášení garanta projektu

Tato práce vznikla za podpory projektu CZ.1.05/2.1.00/03.094: Regionální inovační centrum elektrotechniky (RICE) a projektů TA ČR: TA01010863, CK TE01020038 a TE02000103 (CIDAM), jejichž jsem řešitelem. Potvrzuji, že Ing. Tomáš Košan je hlavním autorem částí, které jsou představeny v této práci.

V Plzni, 10. listopadu 2014

.....  
prof. Ing. Zdeněk Peroutka, Ph.D.

## Poděkování

Na tomto místě bych chtěl poděkovat všem, kteří měli vliv na moji disertační práci. V první řadě mé poděkování patří školiteli panu prof. Ing. Zdeňku Peroutkovi, Ph.D. za odbornou podporu, profesionální poznámky a rady při řešení výzkumných úkolů. Také bych rád poděkoval školiteli specialistovi Ing. Janu Molnárovi, Ph.D. za cenné podněty a odbornou spolupráci. Další díky patří mým kolegům z Katedry elektromechaniky a výkonové elektroniky a Regionálního inovačního centra elektrotechniky. Konečně bych chtěl poděkovat své rodině, že mě vždy podporovala ve studiu a profesní cestě.

Tato dizertační práce vznikla s podporou Evropského fondu pro regionální rozvoj (ERDF) a Ministerstva školství, mládeže a tělovýchovy ČR v rámci projektu CZ.1.05/2.1.00/03.0094: Regionální inovační centrum elektrotechniky (RICE), dále s podporou Technologické agentury České republiky v rámci projektů TA01010863, CK TE01020038, TE02000103 (CIDAM) a studentského grantového systému SGS-2012-071.

V Plzni, 10. listopadu 2014

.....  
Ing. Tomáš Košan



## Anotace

Tato dizertační práce se zabývá zejména problematikou řízení víceúrovňových měničů a využitím hradlových polí pro výpočetně náročné algoritmy pro regulaci pohonů. Realizace těchto úloh je úzce svázána s vhodným řídicím hardware - mikroprocesorovým (číslicovým) regulátorem.

V první části práce jsou rozebrány požadavky na řídicí hardware pro víceúrovňové měniče. Tato část také hodnotí klíčové parametry pro využití hradlových polí pro akceleraci výpočtů regulátorů.

Navazující část práce se zabývá problematikou návrhu mikroprocesorového regulátoru, který by především umožnil přímé připojení víceúrovňových měničů bez dalších hardwarových prostředků. Tato část práce popisuje také další hardwarové prostředky vyvinuté pro zajištění bezproblémového provozu této platformy v prostředí laboratoře výkonové elektroniky. Několik podkapitol se pak věnuje návrhu JTAG emulátorů pro nahrávání a ladění kódu v použitých mikrokontrolérech. Navazuje představení programové podpory pro nahrávání programovatelných logických obvodů pomocí speciální aplikace PLD loader a vyvinutých JTAG emulátorů. Spolu s hardwarem je stručně představeno aplikační rozhraní MLC\_driver\_lib a základní firmware pro použité logické obvody.

Další část práce popisuje dosažené výsledky výzkumu v oblasti víceúrovňových měničů a zejména návrh a implementaci jejich algoritmů řízení. Navazuje část popisující regulační struktury pro jednotlivé topologie ve funkcích usměrňovače a střídače, spolu s výsledky experimentů.

Předposlední část práce prezentuje výsledky výzkumu v oblasti bezsenzorového řízení střídavých elektrických pohonů s využitím stochastických filtrů a bayesovských estimátorů. Tato práce především řeší úlohu implementace marginalizovaného částicového (particle) filtru, určeného pro odhad polohy a rychlosti rotoru synchronního motoru s povrchovými permanentními magnety na rotoru (PMSM) bez použití čidla rychlosti nebo polohy na hřídeli stroje.

Poslední část práce představuje blok pro Matlab/Simulink, který byl vyvinut jako účinný nástroj pro rapid prototyping a plnohodnotný model-based design regulačních struktur výkonových polovodičových měničů a regulovaných pohonů.

## Klíčová slova

Víceúrovňové měniče, programovatelné logické obvody, mikrokontroléry, JTAG, periférie mapované do paměti, FPGA, CPLD, JTAG emulátor, marginalizovaný částicový filtr, Simulink, rapid prototyping

## **Abstract**

This doctoral thesis is focused on the issue of multilevel converter control and using FPGA for computationally intensive algorithms for motor control. The implementation of these tasks is closely related to an appropriate control hardware - the microprocessor (digital) controller.

In the first part of the thesis, requirements for the multilevel converters control hardware are discussed. Key parameters for the use of FPGAs for accelerating calculations of control algorithms are also evaluated in this section.

The following part of the work deals with the proposal of the microprocessor controller, which would enable a direct connection of multilevel converters without additional hardware resources. This part also describes an additional hardware resources developed to ensure a smooth operation of the platform in a power electronics laboratory.

Several subchapters are devoted to JTAG emulators dedicated to loading and debugging code in the microcontroller. A software support for programming of the logic circuits and the developed JTAG emulators using a special application called PLD loader is presented in this section. It is briefly introduced a basic firmware and application interface called `MLC_driver_lib`.

The next part of the work describes achieved results of research into multilevel converters field, especially into design and implementation of control algorithms. Following part describes control structures for all converter topologies used in the research and shows experimental results.

The last but one part of the thesis presents the results of research into the field of sensorless control of AC drives using stochastic filters and Bayesian estimators. This work primarily addresses the role of implementation marginalized particle filter intended to the estimation of rotor position and speed of surface magnet mounted synchronous motors without any sensors at the rotor shaft.

The last part presents the block for Matlab/Simulink which was developed as an effective tool for rapid prototyping and full model-based design of both the power electronics converters and the adjustable speed drives control structures.

## **Keywords**

Multi-level converters, programmable logic devices, microcontrollers, JTAG, memory mapped devices, FPGA, CPLD, JTAG emulator, marginalized particle filter, Simulink, rapid prototyping

## **Annotation**

Diese Doktorarbeit beschäftigt sich vor allem mit der Problematik der Steuerung mehrstufiger Spannungsumrichter und mit einer FPGA-Anwendung für berechnete anspruchsvolle Algorithmen für eine Regulierung der Antriebe. Die Realisierung dieser Aufgaben ist eng mit geeigneter Steuerungshardware - mit (digitalem) Mikroprozessorregler verbinden.

Der erste Teil der Arbeit behandelt die Anforderungen der Steuerungshardware für mehrstufige Spannungsumrichter. In diesem Abschnitt sind auch die Hauptparameter für FPGA-Anwendung beurteilt, um die von Reglern durchgeführte Berechnungen zu beschleunigen.

Der weitere Teil der Doktorarbeit beschäftigt sich mit einem Entwurf des Mikroprozessorreglers, der vor allem eine direkte Verbindung der mehrstufigen Spannungsumrichter ohne Anwendung weiterer Hardwaremittel ermöglicht. Dieser Teil der Arbeit beschreibt auch andere Hardwaremittel die für die Sicherung des problemlosen Betrieb dieser Plattform in der Umgebung der Labor für Leistungselektronik entwickelt wurde. Mehrere Unterkapitel widmen sich dem Entwurf der JTAG Emulatoren , die als Programmlader und Programmdebugger für den benutzten Mikroregler dienen. Dann folgt eine Vorstellung der programmatischen Unterstützung für Programmierung von programmierbaren logischen Schaltungen durch der Spezialanwendung "PLD loader" und entwickelte JTAG Emulatoren. Zusammen mit Hardware ist kurz die Anwendungsschnittstelle MLC\_driver\_lib und die Grund-Firmware für angewandte logische Schaltungen vorgestellt.

Der weitere Teil der Arbeit beschreibt erreichte Ergebnisse der Forschung im Gebiet mehrstufiger Spannungsumrichter und vor allem den Entwurf und die Implementierung ihrer Steueralgorithmen. Dann folgt der Steuerungsstrukturen für einzelne Gleichrichter- und Wechselrichtertopologien beschreibt, zusammen mit Ergebnisse des Experimentes.

Der vorletzte Teil der Arbeit präsentiert die Ergebnisse der Forschung im Gebiet sensorlose Steuerung von elektrischen Wechselstromantrieben mit der Verwendung stochastischer Filter und Bayes-Schätzer. Diese Arbeit löst vor allem die Aufgabe der Implementierung marginalisiertes Partikelfilters, bestimmt für die Lage- und Rotorgeschwindigkeitschätzung des Synchronmotors mit oberflächen permanenten Magneten auf Rotor ohne Verwendung des Geschwindigkeits- oder Lagesensors auf der Welle der Maschine.

Der letzte Teil der Arbeit präsentiert einen Block für Matlab/Simulink, der als effektives Instrument für Rapid Prototyping und vollwertiges Model-Based Design der Steuerungsstrukturen leistungshalbleiterer Spannungsumrichter und regulierenden Antriebe entwickelt wurde.

## **Schlüsselwort**

Mehrstufige Spannungsumrichter, programmierbare logische Schaltungen, Mikroregler, JTAG, in den Speicher abgebildete Peripherien, FPGA, CPLD, JTAG Emulator, marginalisiertes Partikelfilter, Simulink, Rapid Prototyping

# Obsah

<b>Seznam symbolů a zkratk</b>	<b>4</b>
<b>1. Úvod</b>	<b>8</b>
1.1. Současný stav ve zkoumané problematice . . . . .	8
1.2. Cíle práce . . . . .	11
1.3. Metodika řešení . . . . .	12
<b>2. Požadavky na mikroprocesorový regulátor pro řízení vícehladinových měničů a pokročilé řízení střídavých pohonů</b>	<b>14</b>
2.1. Požadavky na hardware pro řízení zkoumaných víceúrovňových topologií měničů . . . . .	14
2.2. Požadavky pro speciální algoritmy řízení . . . . .	17
2.3. Požadavky na technologii výroby . . . . .	18
2.4. Dostupné topologie propojení mikrokontroléru a programovatelných logických obvodů . . . . .	19
2.5. Vývojové a ladicí prostředky pro číslicové regulátory . . . . .	20
2.6. Shrnutí parametrů a požadavků pro realizaci číslicového regulátoru . . . . .	20
<b>3. Vývojová platforma pro implementaci řízení vícehladinových měničů a pokročilé algoritmy regulace střídavých pohonů</b>	<b>22</b>
3.1. Koncepce vývojových nástrojů . . . . .	23
3.2. JTAG emulátory ve výkonové elektronice . . . . .	24
3.2.1. Galvanické oddělení vývojové platformy . . . . .	24
3.3. Emulátory JTAGv3 a JTAGv4 . . . . .	24
3.4. Emulátor JTAGv5 . . . . .	26
3.4.1. Fail-safe funkce emulátoru JTAGv5 . . . . .	28
3.4.2. Nahrávání programovatelných logických obvodů pomocí JTAGv3/4/5 . . . . .	29
3.5. Multi Level Converter Interface - mikroprocesorový regulátor pro víceúrovňové měniče a výpočetně náročné algoritmy regulace v pohonech . . . . .	32
3.5.1. Topologie MLC interface . . . . .	32
3.5.2. Firmware systémového CPLD a jeho vstupy a výstupy . . . . .	35
3.5.3. Základní firmware pro FPGA . . . . .	37
3.5.4. Programové vybavení pro mikrokontrolér - MLC_interface_lib . . . . .	40

3.5.5. Aplikační rozhraní MLC API . . . . .	41
3.6. Moduly pro MLC interface . . . . .	47
3.6.1. Mikrokontrolérové moduly pro MLC interface . . . . .	47
3.6.2. PWM modul pro MLC interface . . . . .	47
3.7. Výsledky měření kritických parametrů MLC interface . . . . .	48
3.7.1. Měřící cyklus analogově-digitálních převodníků . . . . .	49
3.8. Porovnání implementací řízení měniče 4L-FLC z pohledu využitých prostředků	50
3.9. Zhodnocení základní vývojové platformy MLC interface a přidružených nástrojů	52
<b>4. Vývoj PWM modulátorů pro vícehladinové měniče středního a velkého výkonu</b>	<b>54</b>
4.1. Tříúrovňový měnič s upínacími diodami: 3L-NPC . . . . .	54
4.1.1. PWM modulátor pro 3L-NPC . . . . .	55
4.2. Čtyřúrovňový měnič s plovoucími kondenzátory: 4L-FLC . . . . .	55
4.2.1. PD-PWM modulátor pro 4L-FLC . . . . .	58
4.3. Sériově řazené H můstky: 7L-CHB . . . . .	59
4.3.1. PWM modulátor pro 7L-CHB měnič v FPGA . . . . .	60
4.4. Provozování víceúrovňových měničů s omezeným počtem synchronních PWM	62
4.4.1. Návrh nízkourovňového kontroléru pro měniče 4L-FLC a 3L-ANPC . .	62
4.5. Dílčí závěr . . . . .	65
<b>5. Řízení víceúrovňových měničů a komplexních měničových sestav</b>	<b>67</b>
5.1. Tříúrovňový napěťový nepřímý měnič kmitočtu s upínacími diodami . . . . .	67
5.1.1. Regulace napěťového pulzního usměrňovače s topologií 3L-NPC . . . .	67
5.1.2. Regulace synchronního motoru s buzením měničem topologie 3L-NPC	67
5.1.3. Experimentální ověření funkce rekuperačního měniče s upínacími diodami . . . . .	69
5.2. Čtyřúrovňový napěťový nepřímý měnič kmitočtu s plovoucími kondenzátory .	69
5.2.1. Přednabíjení víceúrovňových měničů s plovoucími kondenzátory . . .	69
5.2.2. Regulace napěťového pulzního usměrňovače s topologií 4L-FLC . . . .	72
5.2.3. Regulace asynchronního motoru pro 4L-FLC . . . . .	73
5.2.4. Experimentální ověření přednabíjecí procedury pro 4L-FLC . . . . .	74
5.2.5. Experimentální ověření funkce rekuperačního měniče s plovoucími kondenzátory . . . . .	75
5.3. Jednofázový sedmiúrovňový napěťový usměrňovač složený z H-můstků . . . .	78
5.3.1. Jednofázová synchronizace pro napěťový pulzní usměrňovač . . . . .	78
5.3.2. Regulační schéma pro jednofázový 7L-CHB měnič . . . . .	81
5.3.3. Experimentální ověření funkce napěťového pulzního usměrňovače s topologií 7L-CHB . . . . .	84
5.4. Dílčí závěr . . . . .	86

<b>6. Výpočetně náročné algoritmy v hradlových polích</b>	<b>90</b>
6.1. Návrh marginalizovaného částicového filtru pro PMSM . . . . .	91
6.1.1. Matematický model PMSM . . . . .	92
6.1.2. Marginalizovaný částicový filtr . . . . .	93
6.1.3. Marginalizovaný částicový filtr pro PMSM . . . . .	94
6.1.4. Implementace MPF do FPGA . . . . .	95
6.1.5. Experimentální výsledky . . . . .	97
6.2. Dílčí závěr . . . . .	99
<b>7. Podpora rapid prototypingu pro MLC interface - Model Based Design (MBD)</b>	<b>100</b>
7.1. Blok MLC interface pro Simulink . . . . .	100
7.1.1. Experimentální ověření vyvinutého bloku pro Simulink . . . . .	105
7.2. Dílčí závěr . . . . .	107
<b>8. Závěr</b>	<b>108</b>
8.1. Hlavní přínos práce . . . . .	108
8.2. Perspektivní směry dalšího výzkumu . . . . .	110
<b>Literatura</b>	<b>112</b>
<b>Seznam autorových publikací</b>	<b>120</b>
<b>A. Navržený hardware</b>	<b>126</b>
<b>B. Měřicí pracoviště s víceúrovňovými měniči</b>	<b>130</b>
<b>C. Parametry použitých motorů</b>	<b>132</b>

## Seznam symbolů a zkratek

FLC měnič	měnič s plovoucími kondenzátory
ANPC měnič	měnič s aktivním upínáním
NPC měnič	měnič s upínacími diodami
CHB měnič	sériově řazené H-můstky
$a_d, a_q, b_d, b_q, c_d, c_q, f_q$	konstanty PMSM modelu
$\alpha$ - $\beta$	stojící souřadný systém
BSDL	Boundary Scan Description Language
CCS	Code Composer Studio
CPLD	Complex Programmable Device
$d$ - $q$	rotující souřadný systém
DPS	deska plošného spoje
DSP	Digitální signálový procesor
$\Delta t$	perioda vzorkování
$\Delta u_u$	rozdíl vzorků vstupního napětí
EMI	elektromagnetická interference
$\epsilon_{d,t}, \epsilon_{q,t}, \epsilon_{\omega,t}, \epsilon_{\theta,t}$	šumové složky modelu PMSM
$\epsilon_{estim}$	vypočtený posuv napětí mezi napětím sítě a usměřovače
$f$	frekvence
FPGA	hradlové pole
$\mathbf{I}, \bar{I}$	fázor proudu
$i_{d,t}, i_{q,t}, i_{d,t+1}, i_{q,t+1}$	statorové proudy v souřadném systému $d$ - $q$
$i_{ui}, i_{vi}, i_{wi}$	fázový proud střídače
$I_m$	velikost amplitudy fázového proudu
$i_{ur}, i_{vr}, i_{wr}$	fázový proud usměřovače
$i_{sd}, i_{sq}$	statorové proudy v souřadném systému $d$ - $q$
JTAG	Joint Test Action Group

$L$	indukčnost
$L_{sd}, L_{sq}$	statorové indukčnosti v souřadném systému $d$ - $q$
LVDS	low-voltage differential signaling
LVTTL	low-voltage transistor-transistor logic
MCU	mikrokontrolér
MKO	Monostabilní klopný obvod
MLC interface	Multi Level Converter interface
MPF	marginalizovaný částicový filtr
$N_{2L}$	počet čidel pro dvouúrovňový měnič
$N_{ADC}$	počet A/D převodníků
$N_{clamp}$	počet PWM výstupů pro aktivní clamping
$n_{eff}$	efektivní velikost vzorku
$N_f$	počet fází měniče
$N_{HB}$	počet H-můstků na fázi
$N_{PWM}$	počet PWM signálů
$N_P$	počet tiků za periodu PWM generátoru
$N_L$	počet úrovní měniče
$N_{samples}$	počet vzorků
$\omega$	úhlová rychlost
$\omega_{PLL}$	úhlová rychlost sítě ze SOGI PLL
$\omega_N$	jmenovitá úhlová rychlost sítě pro SOGI PLL
$\hat{\omega}_e$	odhadovaná elektrická rychlost rotoru
$\omega_e^{(i_{best})}$	nejpravděpodobnější odhadovaná elektrická rychlost rotoru
PC	osobní počítač
PCB	deska plošného spoje
PD-PWM	pulzně šířková modulace s amplitudově posunutými pilami
$P_{Hx}$	Výkon dodávaný do zátěže H můstkem s indexem $x$
PLD	hradlové pole
PMSM	synchronní motor s permanentními magnety
PS-PWM	pulzně šířková modulace s fázově posunutými pilami
$q_\omega, q_\vartheta, r$	kovarianční matice
$T_{ADC}$	doba pro získání analogových hodnot



$T_{CONV}$	doba převodu A/D převodníku
TCK	test clock
TDI	test data in
TDO	test data out
TMS	test mode select
TRST	test reset
R	odpor
REMCS	RICE Embedded Modular Control System
RTCK	return test clock
SCI	sériový port
SerDes	sérializátor/desérializátor
$s_{i_u}, s_{i_v}, s_{i_w}$	příznak znaménka proudu pro balancovací tabulku
SoC	system na čipu
SOGI	Second Order Generalized Integrator
SRST	systemový reset
SVF	Serial Vector Format
SV PWM	vektorová pulzně šířková modulace
$t$	čas
$T_{BUS}$	délka cyklu externí datové a adresové sběrnice
$T_{int}$	latence přerušovací rutiny
$T_{PD}$	dopravní zpoždění
$T_{vz}$	perioda vzorkování
$T_x, \overline{T_x}$	komplementární dvojice tranzistorů, index $x$ je pak $i$ pro střídač a $r$ pro usměrňovač
$\vartheta_{HB}$	fázový posuv (přesazení) pily mezi jednotlivými H-můstky
$\vartheta_{PLL}, \vartheta$	poloha vektoru napětí sítě v $\alpha - \beta$
$\hat{\vartheta}_e$	odhadovaná poloha rotoru
$\vartheta_e^{(i_{best})}$	nejpravděpodobnější odhadovaná poloha rotoru
$u$	jako index označuje vazbu na fázi $u$
$U_{cu1,2}, U_{cv1,2}, U_{cw1,2}$	příznaky úrovně nabití plovoucích kondenzátorů pro balancovací tabulku
$U, \overline{U}$	fázor napětí

$U_{cw}$	požadované napětí v meziobvodu
$U_{dc}$	napětí meziobvodu
$U_{Xd c Y}$	napětí meziobvodu, $X$ je označení fáze (u, v, w) a $Y$ je index meziobvodu u H-můstků
USB	Universal Serial Bus
$\bar{U}_{ur}, \bar{U}_{vr}, \bar{U}_{wr}$	fázory fázových napětí usměrňovače
$\bar{U}_m$	velikost amplitudy napětí
$u_\alpha, u_\beta$	napětí v souřadném systému $\alpha - \beta$
$u_{ur}, u_{vr}, u_{wr}$	fázové napětí usměrňovače (vůči virtuální nule)
$u_{ui}, u_{vi}, u_{wi}$	fázové napětí střídače (vůči virtuální nule)
$u_\alpha, u_\beta$	napětí v souřadném systému $\alpha - \beta$
$u_d, u_q$	napětí v souřadném systému $d - q$
$u_{PR}$	výstupní napětí rezonančního regulátoru
$u_{s\alpha}, u_{s\beta}$	rekonstruované napětí statoru v $\alpha - \beta$
$v$	jako index označuje vazbu na fázi $v$
$w$	jako index označuje vazbu na fázi $w$
$w_t$	normalizovaná váha částice
$\hat{w}_t$	nenormalizovaná váha částice
$w_0^{(i)}$	počáteční váha $i$ -té částice
$X_L$	induktivní reaktance
$x_t^n$	nelineární část modelu
$x_t^l$	lineární část modelu

# 1. Úvod

Moderní vysokonapěťové měniče středního a velkého výkonu jsou řešeny jako vícehladinové topologie provozující velký počet spínacích prvků. Toto klade velké nároky na řídicí hardware jak z hlediska disponibilních periférií, tak i z hlediska výpočetních nároků kladených na řídicí obvody, včetně modulátoru. Jedním z cílů této práce je návrh číslicového regulátoru vhodného pro tyto typy úloh. Další cílovou oblastí pro tuto dizertační práci jsou pokročilé algoritmy řízení a identifikace parametrů regulovaných elektrických pohonů, především pak střídavých, které využívají moderní stochastické filtry a bayesovské estimátory a které jsou z hlediska v průmyslu běžně používaného hardwaru extrémně výpočetně náročné.

Tato práce se konkrétně zabývá návrhem, vývojem a testováním hardwaru a algoritmů řízení víceúrovňových měničů a extrémně výpočetně náročných algoritmů, počínaje návrhem specializované řídicí platformy, která na sobě nese veškerý hardware nutný pro provoz víceúrovňových měničů až do sedmiúrovňové varianty, přes vývoj podpůrných nástrojů pro ladění a nahrávání programového vybavení (algoritmů, firmware atp.), konče realizací řízení tříúrovňového měniče s upínacími diodami (NPC), čtyřúrovňového měniče s plovoucími kondenzátory (FLC) a sedmiúrovňového měniče ze sériově řazených můstků (CHB).

Navržený mikroprocesorový (číslíkový) regulátor zároveň umožňuje implementaci a testování výpočetně extrémně náročných algoritmů pro bezsensorová řízení střídavých regulovaných pohonů a obecně náročných paralelně pracujících algoritmů, které by se obtížně realizovaly v samostatném mikrokontroléru z důvodu limitovaného výpočetního výkonu. Na příkladu marginalizovaných částicových filtrů je takováto úloha prakticky ukázána.

## 1.1. Současný stav ve zkoumané problematice

Tato práce, jak bylo vysvětleno v úvodu, se zaměřuje na dvě hlavní cílové oblasti:

- 1/ řízení vícehladinových měničů středního a velkého výkonu
- 2/ výpočetně náročné pokročilé řízení a identifikace parametrů střídavých elektrických pohonů.

V tomto smyslu je též rozdělena analýza současného poznání ve zkoumaných problematikách ad 1/ a ad 2/ níže. Problematika řízení víceúrovňových měničů je dnes již široce zpracována, vzhledem k dostupným technologiím umožňujícím praktickou realizaci jejich komplikovaného řízení.

Víceúrovňové měniče [1, 2, 3] jsou obecně výhodné pro vysokonapěťové aplikace, kdy rozložením napětí na více sériově řazených součástek lze dosáhnout násobného napětí v meziobvodu a zároveň má výsledný proud i napětí menší podíl harmonického zkreslení než běžný dvojúrovňový měnič, a to vlivem tvaru výstupního napětí, které je rozděleno na více hladin daných topologií měniče. Víceúrovňové měniče se dále vyznačují menšími hodnotami  $\frac{du}{dt}$  výstupního napětí (opět vlivem jeho rozložení na několik hladin), což snižuje úroveň emitovaných rušivých signálů. Zároveň nám větší dovolené napětí meziobvodu umožňuje připojit měnič přímo k rozvodné síti bez použití snižovacího transformátoru, to vede k finančním i rozměrovým úsporám. Komplikací použití takovýchto topologií měničů je nutnost balancovat napětí na kondenzátorech které udržují napětí příslušných hladin a tím i napětí blokované jednotlivými součástkami. Problematika balancování výrazně ovlivňuje především používání měničů s plovoucími kondenzátory. Existují topologie víceúrovňových měničů které mají přirozené samobalancování již z principu zapojení [4], případně jednodušší varianta [5]. Tyto topologie se vyznačují složitým zapojením a vysokým počtem řízených prvků, tato práce se zaměřuje na topologie s aktivním balancováním, které jsou obvodově jednodušší. I u těch lze použitím specifického řídicího algoritmu dosáhnout samobalancování (tj. bez zásahu algoritmu), avšak za cenu horších dynamických parametrů, podrobněji viz práce [6, 7, 8, 9]. Použitím aktivního balancování pak lze dosáhnout dobrého dynamického chování spolu s vyváženým napětím na kondenzátorech.

Z implementačního hlediska je třeba se zabývat především návrhem PWM modulátoru, běžně dostupné procesory nedisponují velkým počtem PWM a také jsou obtížně realizovatelné některé algoritmy balancování napětí např. [10], které vyžadují další zásah do hardwarově vytvořené spínací kombinace.

Z pohledu implementace řídicích algoritmů [11] je třeba mít výkonný hardware, který umožní výpočetně náročné algoritmy testovat v laboratoři v systému reálného času. Z nabídky hlavních výrobců [12, 13] vyplývá, že komerčně dostupné vývojové kity s FPGA jsou zaměřené především na implementaci multimediálních úloh. Většina z nich navíc neintegruje procesor vhodný pro použití ve výkonové elektronice. Vývojové kity s mikrokontroléry disponujícími periferiemi vhodnými pro vývoj algoritmů pro řízení měničů a motorů jsou dostupné od renomovaných výrobců [14, 15, 16]. Tyto vývojové kity jsou určeny pro ovládání maximálně dvou třífázových dvouúrovňových měničů (kombinace pulzní usměrňovač a střídač) nebo maximálně jednoho tříúrovňového měniče (střídač nebo usměrňovač). Dále existují na naše požadavky málo vhodná řešení jako např. [17], kde je použitý procesor nevhodný pro potřeby vývoje regulačních algoritmů pracujících v reálném čase, především kvůli komplexní vnitřní struktuře procesoru. Samotný návrh řídicích karet založených na této technologii je komplexní problém (nízká napětí v kombinaci s vysokými rychlostmi). Výhodou takového sestavy je vysoký výpočetní výkon a možnost případně provozovat operační systém typu Linux nebo Windows a v něm např. řídicí systém typu REX [18]. Zde však můžeme narazit na pro-

blémy např. s predikovatelností odezvy systému a velikostí latence mezi přerušením a jeho obsluhou, limitující je i časová granularita systému, tj. nejmenší perioda systémového časovače. Ta se v případě systémů Windows pohybuje v řádu stovek mikrosekund až jednotek milisekund. Obdobná situace je i u operačních systémů založených na Linuxovém monolitickém jádře, zde je však k dispozici real-time rozšíření Linuxového jádra a lze tak dosáhnout latencí nižších, i když v porovnání se systémem bez operačního systému stále násobně větších. Byly již také navrženy systémy pro jednoduše či omezená použití jako jsou [19, 20, 21, 22, 23]. Jednotlivá řešení se liší především možnostmi jiného využití než k účelům ve výše zmíněných článcích. Zásadním cílem při návrhu nového vývojového mikroprocesorového regulátoru jsou cílové aplikace pro jeho využití: řízení nejen víceúrovňových měničů, ale také provozování výpočetně náročných řídicích algoritmů, a regulace komplexních regulovaných pohonů, včetně diagnostiky a komunikace s nadřazenou jednotkou.

Druhý směr výzkumu se zaměřuje na využití hradlových polí pro akceleraci výpočetně náročných algoritmů. Hradlové pole lze využít jako obecně použitelný koprocessor, např. pro operace s maticemi (viz např. práce [24]), kdy operace, které spotřebovávají příliš mnoho výpočetního času mikrokontroléru, navrhne v hradlovém poli. Výpočet se pak rozdělí na dílčí části, některé počítané v mikroprocesoru a jiné v hradlovém poli. Výpočty v obou obvodech mohou běžet zároveň, je tak dosaženo dalšího zvýšení výpočetního výkonu.

V oblasti moderních metod řízení a identifikace hrají významnou roli Kalmanovy filtry (EKF, CDKF, atd.) a lineárně kvadratické (LQ, LQG) řízení. U těchto metod vycházejících z klasické stavové interpretace systémů lze za pomoci matematického koprocessoru akcelerujícího operace s maticemi dosáhnout velmi krátkých výpočetních časů, jež jsou předpokladem pro jejich uplatnění v oblasti řízení elektrických pohonů a výkonových měničů.

Mezi další nastupující trendy v oblasti řízení patří metody vycházející z Monte Carla. Mezi tyto metody, které umožňují identifikaci skrytých stavových veličin (i s jiným než Gaussovským rozdělením) patří částicové (particle) filtry. Tyto filtry (viz např. [25, 26, 27]) jsou obecně výpočetně náročné, zejména paralelním výpočtem matematických modelů (částic) počítaných pro různé stavy. Podobný princip z pohledu algoritmů využívají prediktivní metody řízení s konečnou množinou akčních zásahů tzv. Finite-Set Model Predictive Control (FS-MPC) (např. [28, 29, 30, 31]). I zde jsou počítány paralelně matematické modely řízených systémů se shodným stavem, ale s rozdílným akčním zásahem. Využití hradlových polí pro real-time aplikace těchto algoritmů v krátkých časových smyčkách bývá nezbytné.

Dalšími metodami identifikace a řízení z pohledu algoritmů jsou hybridní metody kombinující složité maticové operace a paralelní výpočet matematických modelů. Mezi tyto metody z oblasti identifikace lze zařadit např. Unscented Kalman Filtr (UKF) [32], Extended Kalman Particle filtr [33], Unscented Kalman Particle filtr [34] a jiné. Z oblasti řízení jsou to např. metody kombinující konečnou množinu akčních zásahů FS-MPC a LQ řízení pro predikci na dlouhém horizontu.

Z výše uvedených metod tato práce ukazuje použití marginalizovaných částicových filtrů v kombinaci s Kalmanovým filtrem pro estimaci polohy synchronních motorů typu PMSM do hradlového pole [35].

## 1.2. Cíle práce

Na základě posouzení současného stavu poznání ve zkoumané problematice a ve vazbě na autorovo působení při řešení vědecko-výzkumných a průmyslových projektů byly stanoveny následující cíle této dizertační práce:

1. Navrhnout komplexní hardwarové řešení - mikroprocesorový regulátor, který bude umožňovat řízení výkonového polovodičového měniče s velkým počtem spínacích prvků (až sedmiúrovňová topologie měniče) a zároveň bude disponovat s dostatečně velkým výpočetním výkonem tak, aby v něm bylo možné implementovat a zkoumat extrémně výpočetně náročné algoritmy řízení a identifikace parametrů měničů a střídavých elektrických pohonů.
2. Navrhnout spolehlivý, EMI odolný, univerzální JTAG emulátor pro použití v laboratorích výkonové elektroniky, který bude podporovat real-time ladění, programování, vyčítání paměti dat atd. pro vybrané mikrokontroléry.
3. Vytvořit kompletní programovou podporu pro nový regulátor (firmware pro programovatelné obvody, ovladače/knihovny pro mikrokontrolér, příp. aplikace pro PC) tak, aby uživatel nemusel mít podrobné znalosti o hardwaru samotném a mohl se plně soustředit na implementaci řešeného úkolu.
4. Navrhnout, implementovat a otestovat vybrané metody řízení vícehladinového měniče s plovoucími kondenzátory v ustálených a přechodových stavech.
5. Navrhnout, implementovat a otestovat vybrané metody řízení vícehladinového měniče s upínacími diodami v ustálených a přechodových stavech.
6. Navrhnout, implementovat a otestovat vybrané metody řízení měničů založených na topologii sériově řazených H můstků.
7. Navrhnout, implementovat a otestovat vybrané výpočetně náročné řídicí algoritmy střídavých regulovaných pohonů.
8. Návrh a vývoj nástrojů pro rapid prototyping a podporu plnohodnotného model-based designu náročných algoritmů řízení střídavých regulovaných pohonů a složitých měničových sestav v prostředí MATLAB/Simulink.

### 1.3. Metodika řešení

Pro dosažení vytyčených cílů je nutné postupovat od základních stavebních kamenů, jako je teoretický návrh požadavků na hardware regulátoru, přes návrh základního programového vybavení, základních regulačních struktur a specializovaných modulátorů, až po kompletní řešení řízení měničových sestav a výpočetně náročných algoritmů řízení střídavých regulovaných pohonů. Práce je rozdělena do několika hlavních kapitol, každá z nich se zabývá vybranou partií. Komplikované záležitosti jsou rozděleny v rámci kapitol na dílčí sekce.

Vlastní řešení začíná kapitolou 2, která podrobně rozebírá požadavky na hardware, který vyhoví z pohledu použití, realizace, ceny a především vlastnostmi potřebnými pro úkoly vytyčené v cílech práce. V této části jsou podrobně specifikovány požadavky na hardware navrhovaného regulátoru odrážející požadavky aplikací, jejichž algoritmy řízení budou v regulátoru implementovány. Rozhodujícími kritérii pro návrh hardwaru byly zvoleny zejména tyto parametry:

- počet měřících analogově-digitálních vstupů,
- počet digitálních výstupů pro ovládání výkonových prvků,
- samotná architektura vývojového kitu (číslicového regulátoru).

Následující kapitola 3 popisuje realizovanou vývojovou hardwarově-sofwarovou platformu. V první části této kapitoly jsou popsány univerzální programovací a ladicí nástroje, navržené s ohledem na vysokou spolehlivost z hlediska EMC, neboť se předpokládá, že zařízení bude využíváno v prostředí s výrazným elektromagnetickým rušením, které v laboratorních podmínkách násobně překračuje normami povolené EMC limity.

Následuje popis vyvinutého číslicového regulátoru a doplňkového modulu pro rozšíření počtu dostupných PWM výstupů. Tato část práce také diskutuje výsledky měření kritických parametrů navrženého číslicového regulátoru. V dílčím závěru je též zhodnocen přínos vyvinuté platformy v porovnání s předchozí generací vývojové platformy.

V kapitole 4 jsou definovány základní topologie výkonových polovodičových měničů, na které se zaměřil výzkum prezentovaný v této práci, dále jsou zde formulovány hlavní požadavky na řízení těchto měničů, které jsou následně promítnuty do návrhu specializovaných PWM modulátorů.

Navazuje kapitola 5, kde jsou popsány navržené algoritmy řízení zkoumaných měničů, včetně jejich startovacích a vypínacích procedur. Konkrétně je zde popsáno řešení řízení čtyřúrovňového nepřímého měniče kmitočtu, včetně unifikovaného přednabíjecího postupu použitelného jak pro pulzní usměrňovač, tak pro napěťový střídač. Dále je prezentováno navržené řešení tříúrovňového nepřímého měniče kmitočtu s upínacími diodami. Kapitulu uzavírá návrh sedmiúrovňového jednofázového napěťového pulzního usměrňovače založeného na topologii sériově řazených H můstků.

Pro všechny popisované topologie vícehladinových měničů byly navrženy a sestaveny laboratorní prototypy, které byly podrobně testovány a jejich chování analyzováno v řadě přechodových i ustálených stavů, a to jak separátních bloků (pulzní usměrňovač nebo napěťový střídač), tak i kompletní řešení nepřímého frekvenčního měniče s danou topologií.

V kapitole 6 se práce zabývá výpočetně náročnými algoritmy řízení a identifikace parametrů střídavých pohonů využívajících metod statistické filtrace a bayesovské techniky. Dále jsou zde navrženy a řešeny a nástroje pro urychlení vývoje zkoumaných regulačních struktur. Jako vzorový zástupce výpočetně náročných algoritmů byl vybrán marginalizovaný částicový (particle) filtr (MPF) použitý pro odhad polohy a rychlosti rotoru PMSM bez použití čidla na hřídeli stroje. Navržený algoritmus MPF byl implementován do FPGA a byla ozkoušena jeho funkčnost.

Poslední část práce (Kap. 7) řeší a popisuje navržené nástroje pro rapid prototyping a plnohodnotný model-based design pro vyvinutý číslicový regulátor implementované v Matlabu, resp. jeho nadstavbě Simulink. Navržený blok poskytuje uživateli jednoduchou možnost jak začít s regulátorem pracovat. Blok je navržen tak, aby podporoval všechny klíčové funkce regulátoru a zároveň, aby měl v základu předpřipravenou svou strukturu pro jednoduchou implementaci regulačních algoritmů.

Závěr pak shrnuje dosažené výsledky práce, zdůrazňuje hlavní přínos práce a formuluje směry dalšího výzkumu v řešených oblastech.



## 2. Požadavky na mikroprocesorový regulátor pro řízení vícehladinových měničů a pokročilé řízení střídavých pohonů

První část práce je zaměřena na vývoj hardwarové platformy, která bude využita jako číslicový regulátor. Nově vyvinutý regulátor volně navazuje na práci [36], vývojová platforma vyvinutá v rámci této práce jej má postupně nahradit s ohledem na dříve uvedené požadavky, které již stávající hardware není schopen efektivně pokrýt. Částečná zpětná kompatibilita mezi oběma platformami zjednoduší přechod ze starší na novější platformu.

Obě obecné úlohy, řízení samostatného měniče (modulátor) a nadřazené regulační algoritmy, se se svými požadavky potkávají na úrovni koprocesoru připojeného k základnímu mikrokontroléru. Samotné využití tohoto koprocesoru se pak liší dle vyvíjené aplikace a je tedy nutné, aby tento koprocesor byl obecná univerzální součástka. Toto vede na použití programovatelného logického obvodu, pro jednoduché aplikace zastoupené obvodem typu CPLD, pro složitější pak FPGA.

Dalším požadavkem pro návrh hardwaru regulátoru je jeho využití zejména ve vědecko-výzkumných úlohách a v omezeném rozsahu také ve výuce. Tomu by měla odpovídat celková koncepce regulátoru, z pohledu kompatibility se stávajícími technologiemi, jednoduchosti použití a nenáročné údržby (jednoduchá možnost opravy, rozšíření, adaptace na nové úlohy atp.).

### 2.1. Požadavky na hardware pro řízení zkoumaných víceúrovňových topologií měničů

Řízení víceúrovňových měničů vyžaduje především vysoký počet synchronních PWM výstupů. Obecně lze říci, že pro  $N$ -úrovňovou topologii měniče v základním provedení, tj. žádné doplňkové aktivní součástky pro balancování napětí na jednotlivých hladinách, je nutno mít k dispozici  $N_{PWM}$  signálů pro prvky střídače (či napěťového pulzního usměrňovače) dle rovnice (2.1). Speciální případy nejsou brány v potaz, např. H můstky s nerovnoměrným rozložením napětí v meziobvodech jednotlivých buněk, které umožňují dosáhnout stejného počtu úrovní s menším počtem prvků (viz [3]).

$$N_{PWM} = 2.(N_L - 1).N_f \quad (2.1)$$

kde

$N_{PWM}$  ... výsledný počet PWM signálů pro výkonové prvky

$N_f$  ... počet fází měniče

$N_L$  ... počet požadovaných úrovní

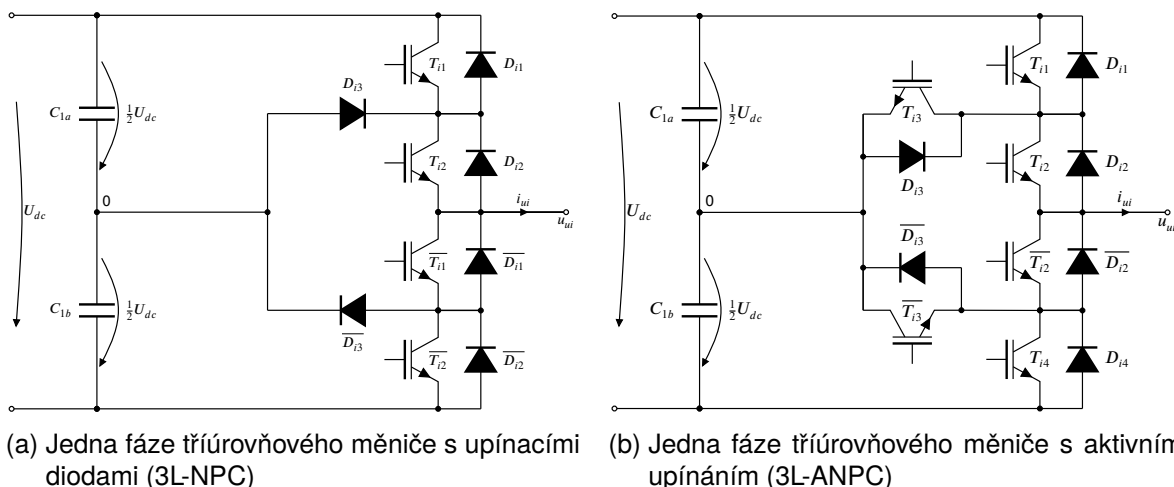
Třífázový tříúrovňový měnič (Obr. 2.1a) s upínacími prvky využívá 12 prvků, v případě aktivního upínání (Obr. 2.1b) potom množství spínacích prvků, a tím i potřebných PWM signálů, narůstá o hodnotu 6. Řízení kompletního nepřímého frekvenčního měniče sestávajícího ze dvou 3L-NPC měničů vyžaduje 24 řídicích signálů, u varianty 3L-ANPC je potřeba 36 PWM signálů.

Pro čtyřúrovňový měnič s plovoucími kondenzátory (Obr. 2.2a) je nutné použít 18 PWM signálů (třífázová varianta). Pro řízení kompletního rekuperačního provedení frekvenčního měniče složeného ze dvou 4L-FLC měničů je tedy potřeba 36 PWM signálů.

Jednofázová varianta sedmiúrovňového měniče s H můstky (Obr. 2.2b) využívá 12 výkonových prvků, třífázová již 36 a kompletní rekuperační frekvenční měnič vyžaduje 72 řídicích PWM signálů.

Víceúrovňové topologie měničů také vyžadují vyšší počet senzorů pro měření napětí na kondenzátorech na nichž dochází ke stabilizaci jednotlivých úrovní napětí. Počet měřených hodnot se liší nejen v závislosti na počtu úrovní měniče, avšak také závisí na jeho topologii.

Neoptimalizovaný počet čidel pro měření napětí na kondenzátorech měniče s plovoucími kondenzátory lze obecně spočítat pomocí rovnice (2.2). K výsledku je však potřeba přičíst i všechna čidla, která by byla na dvojúrovňovém měniči, tj. u pulzního usměrňovače



Obr. 2.1.: Základní schémata zkoumaných topologií víceúrovňových měničů

měření vstupních napětí, napětí v meziobvodu atd.

$$N_{ADC} = (N_L - 2) \cdot N_f \quad (2.2)$$

kde

- $N_{ADC}$  ... výsledný počet měřených analogových signálů
- $N_L$  ... počet požadovaných úrovní
- $N_f$  ... počet fází měniče

Např. pro námi zkoumanou konfiguraci čtyřúrovňového napěťového nepřímého měniče kmitočtu dostáváme výslednou hodnotu 19 analogových vstupů, pokud je možné měřit proudy jen ve dvou fázích ze tří a se synchronizací NPU jen od dvou fází. Pokud by bylo nutné měřit všechny proudy a všechna vstupní napětí, bude potřeba 22 analogových vstupů.

Situace okolo analogových vstupů je podstatně lepší v případě měniče s upínacími diodami, kdy lze čidla měřící napětí na kondenzátorech sdílet mezi jednotlivými fázemi (pro všechny fáze je jeden společný meziobvod tvořený vhodným počtem sekcí kondenzátorů, na kterých se potom tvoří požadované úrovně napětí). Tím se počet měřených analogových signálů výrazně redukuje (viz rovnice (2.3)).

$$N_{ADC} = N_L - 1 \quad (2.3)$$

kde

- $N_{ADC}$  ... výsledný počet měřených analogových signálů
- $N_L$  ... počet požadovaných úrovní měniče

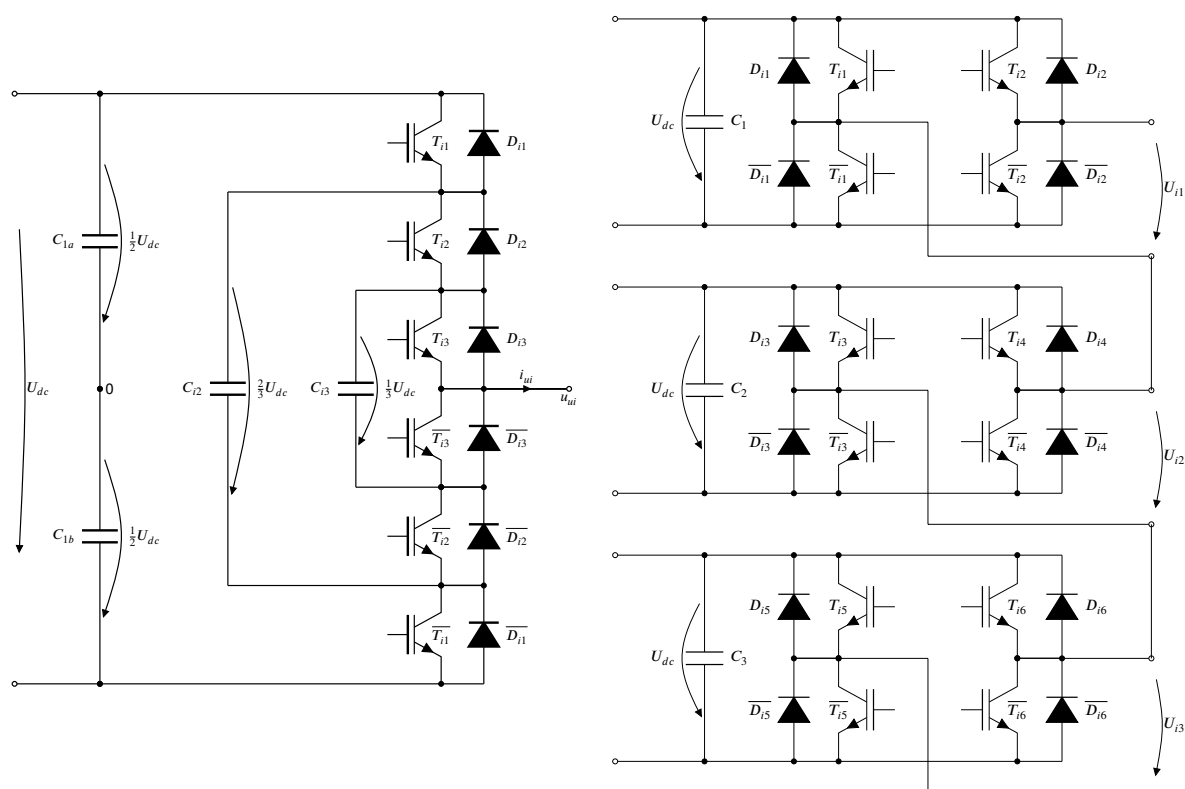
Pokud je uvažován napěťový nepřímý měnič frekvence 3L-(A)NPC, pak je nutných minimálně 10 analogových vstupů pro omezenou variantu a 12, pokud je třeba měřit všechna napětí i proudy.

Poslední topologií měničů, zkoumaných v této práci, jsou sériově řazené H můstky. Obecně lze počet nutných analogových vstupů spočítat dle rovnice (2.4). Ta však platí pouze pro případ symetrických H můstků, tj. u všech máme stejné napětí meziobvodu  $U_{dc}$ .

$$N_{ADC} = (N_L - 1)/2 \quad (2.4)$$

kde

- $N_{ADC}$  ... výsledný počet měřených analogových signálů
- $N_L$  ... počet požadovaných úrovní měniče



(a) Jedna fáze čtyřúrovňového měniče s plovoucími kondenzátory (4L-FLC) (b) Jedna fáze sedmiúrovňového měniče se sériově řazenými H můstkami (7L-CHB)

Obr. 2.2.: Základní schémata zkoumaných topologií víceúrovňových měničů

Požadované počty PWM výstupů pro víceúrovňové měniče naznačují, že se nelze vyhnout použití programovatelného logického obvodu typu FPGA, který disponuje velkým počtem digitálních vstupů/výstupů. Dále požadavky na počet analogových vstupů vedou na použití externího analogově-digitálního převodníku. Externí převodník také řeší problémy spojené s převodníky integrovanými přímo na čipu mikrokontroléru, jako je např. zarušení od běžícího jádra. Tento problém lze sice obejít uspáváním jádra procesoru a přidružených periférií, avšak v případě řízení měniče nelze například vypnout čítače generující PWM signály, což se na výsledcích měření může negativně projevit.

## 2.2. Požadavky pro speciální algoritmy řízení

Realizace speciálních algoritmů řízení klade specifické požadavky na komunikaci mezi ko-procesorem (FPGA) a hlavním řídicím mikrokontrolérem. Komunikace by především měla mít malou latenci a velkou rychlost přenosu, zároveň by při přenosu větších množství dat neměla zbytečně zatěžovat řídicí mikrokontrolér. Pro přenos velkého množství dat lze využít jak sériové, tak paralelní sběrnice [37]. Z pohledu programovatelných obvodů je možné využít obě možnosti, avšak z pohledu mikrokontroléru je výhodnější použít paralelní sběrnice,

konkrétně rozhraní pro připojení vnějších pamětí/periferií. Ta pak umožní mapovat případné periferie v CPLD/FPGA přímo do paměťového prostoru, což by v případě např. sériové SPI sběrnice šlo realizovat obtížně. Paralelní přenos dat může mít obecně vyšší přenosovou rychlost než přenos po sběrnici sériové. Výjimkou jsou ultra rychlé diferenciální sériové sběrnice využívající technologie low-voltage differential signaling (LVDS) [38], ty však vyžadují specializovaný hardware pro převod z LVTTTL úrovní na úrovně používané technologií LVDS a implementaci SerDes (serializer - deserializer) převodníků. V regulační technice běžně využívané mikrokontroléry však hardwarovou podporou pro LVDS komunikaci většinou nedisponují, naopak v FPGA není implementace SerDes převodníku obtížná. LVDS je navíc jednosměrný přenos, tedy pro komunikaci point-to-point je potřeba linek minimálně dvou. Od použití LVDS bylo tedy upuštěno, tato sběrnice má využití např. u rozsáhlých systémů typu Rice Embedded Modular Control System (REMCS) [39], kde se velice dobře uplatní pro rychlou komunikaci mezi jednotlivými kartami.

Nejvýhodněji se z realizačního hlediska jeví paralelní sběrnice mikrokontroléru, která je z jeho pohledu transparentní a její obsluhu lze realizovat i v programovatelných logických obvodech.

### 2.3. Požadavky na technologii výroby

Požadavky na technologii výroby je třeba zohlednit z pohledu zamýšleného použití číslicového regulátoru pro výzkum, vývoj a testování polovodičových měničů a jejich řídicích algoritmů. Je vhodné, aby byl regulátor navržen tak, že bude relativně snadná výměna poškozených částí ručně s pomocí standardních nástrojů (především vstupních a výstupních oddělovačů a přizpůsobovačů úrovní). Tento požadavek téměř vylučuje použití součástek v bezvývodových pouzdrech typu QFN, BGA atd. Vyloučením těchto typů pouzder bude možné navrhnout DPS s menším počtem vrstev ovšem za cenu větších rozměrů DPS.

Z pohledu připojování periferií (měničů a čidel) by mělo být zachované standardní zapojení konektorů aktuálně používaných v našich laboratořích, tj. pinové lišty s roztečí 100 mil. Toto vede na kombinovanou technologii vývodových součástek (konektory) a součástek pro povrchovou montáž ve vývodových pouzdrech např. SO, LQFP.

Klíčové komponenty by měly mít případně několik vývodově kompatibilních variant, aby bylo možné pro méně náročné aplikace regulátor osadit levnějšími součástkami. S tím souvisí nutnost implementovat identifikaci hardwarové varianty z uživatelského programu.

## 2.4. Dostupné topologie propojení mikrokontroléru a programovatelných logických obvodů

Z pohledu dostupných topologií lze pro účely definované v této práci vybrat několik řešení výše uvedených požadavků:

1. Zkombinovat mikrokontrolér s doplňkovou logikou typu CPLD. Tato technologie má omezené použití, neboť CPLD primárně slouží jako náhrada mnoha hradlových obvodů a neumožňuje efektivně implementovat výpočetní bloky. Její využití má smysl tedy pouze v případě, že implementovaná funkčnost není příliš komplexní. Je použitelná např. pro nenáročné vstupy a výstupy, jednoduchou doplňkovou logiku nebo pro zdvojnásobení počtu PWM signálů z mikrokontroléru.
2. Připojit k mikrokontroléru doplňkovou logiku typu FPGA. U této topologie se využijí výhody obou aktivních prvků, mikrokontrolér umožňuje snadné programování a ladění implementovaného algoritmu, zatímco hradlové pole poskytuje masivní paralelizaci výpočetně náročných algoritmů zároveň s velkým počtem uživatelsky konfigurovatelných signálů využitelných u složitých topologií měničů (vícehladinové, sériově či paralelně řazené struktury atd.). Je třeba zmínit dvě krajní varianty propojení mikrokontrolér - FPGA, buď se do FPGA zavedou pouze nezbytně nutné signály z mikrokontroléru, aplikační vývody pak poskytne mikrokontrolér i hradlové pole nezávisle. Nebo se naopak zapojí všechny aktivní vývody mikrokontroléru do FPGA, které pak slouží jako směrovač (router) těchto signálů na aplikační vývody regulátoru. Mezi oběma řešeními je možné plynule přecházet.
3. Implementovat mikrokontrolér přímo v FPGA. Většina výrobců hradlových polí poskytuje také tzv. "soft-core" tj. mikroprocesorová jádra syntetizovatelná do FPGA obvodu. Takto implementovaný mikrokontrolér lze používat jako obdobu čistě hardwarového provedení. Nevýhodou je, že tyto mikrokontroléry nedosahují výkonu reálných mikroprocesorů a navíc alokují zdroje FPGA, je tak k dispozici méně hradel pro realizaci doplňkových funkcí. Typickými představiteli jsou Nios II výrobce Altera [40] či MicroBlaze firmy Xilinx [41].
4. Využít nově dostupné SoC, tedy FPGA s integrovanými hardwarovými mikrokontrolérovými jádry, většinou architektury ARM. V poslední době jsou dostupné čipy typu "System on a Chip", které integrují FPGA spolu s hardwarovým procesorem (většinou architektury ARM) a další periferie běžně se v FPGA nevyskytující. Typickými představiteli jsou Cyclone V SoC firmy Altera [42] či Zynq výrobce Xilinx [43].

Posouzením vlastností jednotlivých topologií byla vybrána varianta 2. Budou použity dva obvody: mikrokontrolér a hradlové pole. Tato varianta byla zvolena především proto, že řešitel-

ské pracoviště má velmi dobrou dlouhodobou zkušenost s mikrokontroléry Texas Instruments řady C2000 a ty se vyrábí pouze jako samostatné obvody, hradlové pole tedy musí být doplněno externě. Vzhledem k požadavku na velký počet PWM výstupů a zároveň na technologii výroby definovanou v sekci 2.3 byla zvolena varianta minimálního propojení mikrokontrolér - FPGA jen s pomocí adresové, datové sběrnice a pouze několika asynchronních signálů, které budou sloužit pro signalizaci (přerušení).

## 2.5. Vývojové a ladicí prostředky pro číslicové regulátory

Aby bylo možné vývojovou platformu plně využít, je třeba mít vývojové prostředky, a to jak software, tak hardware. Vývojový software poskytují výrobci příslušných programovatelných součástek, stejně tak dodávají i hardware nutný pro nahrání a ladění software do cílové platformy ve formě JTAG emulátoru. Alternativně je možné nalézt open source projekty, kterými lze nahradit proprietární software.

Problematické je využití JTAG emulátorů v zarušeném prostředí, v případě našeho okruhu cílových aplikací je problém především souhlasné napětí vznikající při spínání výkonových prvků měniče a EMI rušení generované při provozu frekvenčních měničů, pulzních zdrojů atp. umocněné navíc tím, že se většinou jedná o laboratorní prototypy s otevřenou konstrukcí, které nemají dostatečné elektromagnetické stínění omezující jejich vyzařování. JTAG emulátor, který bude použit pro programování mikrokontroléru, tedy musí mít dostatečnou odolnost vůči oběma výše zmíněným rušivým vlivům. Tím bude zajištěno bezproblémové ladění kódu v mikrokontroléru i za běhu měniče na plném výkonu s provozním napětím v řádu stovek voltů.

## 2.6. Shrnutí parametrů a požadavků pro realizaci číslicového regulátoru

Analýzou výše uvedených informací je možné definovat základní parametry, které má vývojová platforma splňovat:

- Topologie bude kombinovat mikrokontrolér dle našeho výběru spolu s FPGA.
- Budou použity externí analogově-digitální převodníky, tím je omezeno rušení od jádra mikrokontroléru omezující přesnost převodu analogových hodnot.
- Propojení mezi mikrokontrolérem, FPGA a ADC bude využívat paralelní adresovou a datovou sběrnici. Periferie budou z mikrokontroléru dostupné přes paměťový prostor.
- Použité součástky budou výhradně vývodové typy (ne BGA, QFN atp.) včetně těch pro povrchovou montáž.

- Unifikovaná hardwarová podpora programování a ladění kódu v použitých programovatelných obvodech jedním univerzálním JTAG emulátorem.
- Tento JTAG emulátor musí být dostatečně odolný vůči rušení, které generuje měnič při svém chodu.
- Klíčové součástky by měly mít vývodově kompatibilní levnější varianty, které umožní jednoduché škálování výsledné ceny číslicového regulátoru.

Dále je nutné zvážit požadované počty vstupů a výstupů a tomu návrh mikroprocesorového regulátoru dále uzpůsobit. S využitím vztahů z předchozí analýzy počtu nutných PWM signálů z předchozího textu, byly spočteny požadavky pro všechny topologie uvažované pro realizaci. Výsledky jsou shrnuty v Tab. 2.1, kde údaje v závorce platí pro případ, kdy se měří všechny fázové proudy i vstupní napětí v případě NPU. V tabulce červeně zvýrazněné hodnoty budou vzaty jako výchozí minimální požadované počty jednotlivých vstupů či výstupů pro navrhovaný hardware mikroprocesorového regulátoru.

Tab. 2.1.: Analogové vstupy a PWM výstupy pro jednotlivé uvažované topologie měničů

Topologie	Počet PWM	Počet analogových vstupů
4L-FLC frekvenční měnič	36	19 (22)
7L-FLC pulzní usměrňovač	36	<b>21 (23)</b>
3L-NPC frekvenční měnič	24	10 (13)
3L-ANPC frekvenční měnič	36	10 (13)
7L-CHB frekvenční měnič 1f varianta	<b>24</b>	6
7L-CHB frekvenční měnič	<b>72</b>	15 (18)



### 3. Vývojová platforma pro implementaci řízení vícehladinových měničů a pokročilé algoritmy regulace střídavých pohonů

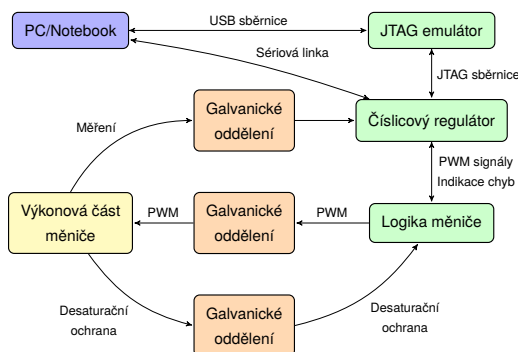
Kapitola je zaměřena na vyvinutý hardware, který slouží k provozování víceúrovňových měničů a zkoumání definovaných výpočetně extrémně náročných algoritmů řízení a identifikace parametrů střídavých regulovaných pohonů, je však vhodný i pro jiné náročné aplikace nejen z oblasti elektrických pohonů.

První část popisuje s JTAG emulátory vyvinuté pro programování a ladění mikrokontrolérů TMS320F2812, TMS320F28335 a Hercules TMS570LS3137. Tyto emulátory jsou založeny na designu JTAG emulátoru od Texas Instruments typu XDS100v1 a v případě JTAGv5 na XDS100v2 viz [44]. Původní návrh byl zásadním způsobem upraven při zachování softwarové kompatibility. Tato kompatibilita s emulátory řady XDS100 umožňuje využít vývojové prostředky pro tyto mikrokontroléry přímo od jejich dodavatele (firmy Texas Instruments), konkrétně se jedná o Code Composer Studio (dále jen CCS).

Oproti původnímu návrhu hardwaru JTAG emulátoru byla provedena některá zjednodušení, ale naopak i podstatná vylepšení a úpravy pro zvýšení odolnosti proti EMI a souhlasným napětím generovaným při provozu měničů.

Následující oddíl kapitoly rozebírá Multi Level Converter interface (dále jen MLC interface), který je speciálně navržen tak, aby:

- Řešil problémy s A/D převodníky u procesorů TMS320F2812 a 28335, které jsou integrovány na čipu, takže výsledky jejich převodu jsou negativně ovlivněny rušením od jádra mikrokontroléru.
- Poskytoval dostatek PWM výstupů a dalších univerzálních vstupů a výstupů se zaměřením na víceúrovňové měniče.
- Umožnil paralelní běh výpočtů např. modelů motorů, estimátorů, částicových (particle) filtrů a podobně.
- Byl co nejvíce kompatibilní s předchozí verzí interface (pro mikrokontroléry TMS320F2812 a TMS320F28335) používaného ve výuce.
- Poskytoval možnost rozšíření funkcionality pomocí externích modulů mapovaných do adresního prostoru řídicího mikrokontroléru.



Obr. 3.1.: Typické blokové schéma vývojového pracoviště

Poslední dvě podkapitoly pojednávají o nízkourovňových kontrolérech s CPLD, které slouží k rozšíření počtu PWM u procesorů TMS320F2812 a 28335 tak, aby bylo možno pracovat s víceúrovňovými měniči.

### 3.1. Koncepce vývojových nástrojů

Typická sestava hardwaru při vývoji je složena z nadřazeného počítače, notebooku nebo pracovní stanice propojené s hardwarem, pro který je vyvíjen software (algoritmus) pomocí JTAG emulátoru. Ten může být připojen na různé externí sběrnice, dříve se využíval paralelní port, nyní výhradně modernější USB sběrnice, případně Ethernet.

Paralelní port v současnosti nebývá standardním IO rozhraním osobních počítačů. Jeho použití je pro moderní vývojový nástroj nevhodné. Základní deska počítače jej sice většinou má implementovaný v rámci zpětné kompatibility, avšak často není vyveden na z vnějšku dostupný konektor. U současně dostupných modelů notebooků je paralelní port dostupný pouze pomocí dokovací stanice nebo adaptérem z PCMCIA nebo Express Card sběrnice.

Naopak USB sběrnice je široce rozšířená a podporovaná na všem hardwarových platformách. Vývoj JTAG emulátoru byl zaměřena pouze na USB variantu, i za předpokladu, že obvodový návrh bude složitější a méně odolný vůči vnějšímu rušení.

Na obr. 3.1 je naznačena typická konfigurace pracoviště pro vývoj algoritmů pro výkonovou elektroniku. Je zde nadřazená jednotka zastoupená osobním počítačem, JTAG emulátor pro programování mikroprocesorového regulátoru a ladění programu, který pak pomocí případné doplňkové logiky zajišťuje řízení měniče, resp. jeho výkonových prvků. Z měniče pak do řídicí hardwarové platformy vstupují chybové signály a signály ze sensorů. Množství těchto signálů závisí na vyvíjené aplikaci, jak bylo uvedeno výše. Komunikace zajišťující ovládání číslicového regulátoru nadřazenou řídicí a diagnostickou jednotkou je realizována obvykle pomocí některého typu sériové linky, např. SCI nebo spolehlivější CAN.

Tato základní konfigurace má však zásadní nedostatek, kdy je nadřazená jednotka (PC/Laptop) galvanicky spojená s mikroprocesorovým regulátorem. Toto propojení pak umožňuje průchod souhlasného napětí vznikajícího při práci výkonové části měniče přes JTAG a následně USB sběrnici. Tato forma rušivého napětí nepříznivě ovlivňuje spolehlivost komunikace s JTAG emulátorem i komunikaci po sériové lince. Možná řešení jsou naznačena dále.

## 3.2. JTAG emulátory ve výkonové elektronice

Používání JTAG emulátorů (a obecně jakékoli komunikace) připojených pomocí USB sběrnice společně s laboratorními prototypy měničů klade na USB zařízení velké nároky z hlediska odolnosti proti indukovanému rušení a souhlasnému napětí vznikajícím při přepínání výkonových prvků. Cílem této práce není tyto jevy redukovat na úrovni měniče či jeho řízení, ale JTAG emulátor vyvinout tak, aby byl proti těmto rušivým vlivům maximálně odolný.

Problém indukovaného EMI lze omezit pečlivým návrhem DPS emulátoru v kombinaci se stíněným kabelem JTAG sběrnice, případně minimalizací délky nestíněné části této sběrnice. Souhlasné napětí měniče lze částečně potlačit důsledným galvanickým oddělením nadřazené jednotky (PC/Laptop) od mikroprocesorového regulátoru.

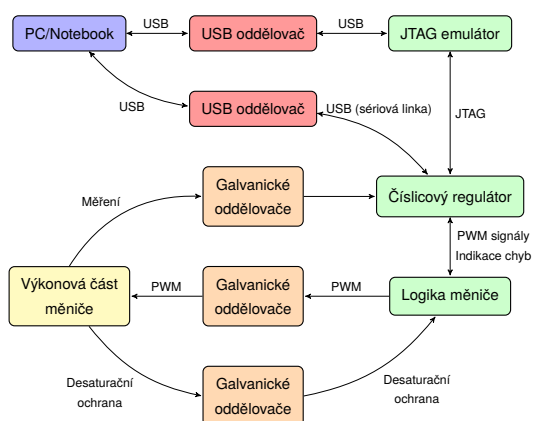
### 3.2.1. Galvanické oddělení vývojové platformy

Galvanické oddělení pro potlačení vlivu souhlasného napětí měniče lze realizovat dvěma způsoby. Pomocí speciálního obvodu je možné oddělit USB sběrnici nebo lze galvanicky oddělit jednotlivé signály JTAG sběrnice. První způsob je naznačen na Obr. 3.2. Navržené zařízení je znázorněno na Obr. 3.3). Toto zařízení používá katalogová zapojení obvodů ADuM5000 [45] a ADuM3160 [46]. Toto řešení je vhodné tam, kde oddělená USB sběrnice není k dispozici a již není možné provést dodatečnou úpravu připojeného hardwaru, který je třeba galvanicky oddělit.

Principiálně je druhá varianta, kdy oddělíme signály JTAG sběrnice, naznačena na Obr. 3.4. USB zařízení, respektive JTAG emulátor je připojen přímo na USB sběrnici, oddělovače jsou vloženy do cesty signálům JTAG sběrnice. Navržené zařízení je zobrazeno na Obr. 3.5. Následující kapitola podrobněji rozebírá takto implementované emulátory.

## 3.3. Emulátory JTAGv3 a JTAGv4

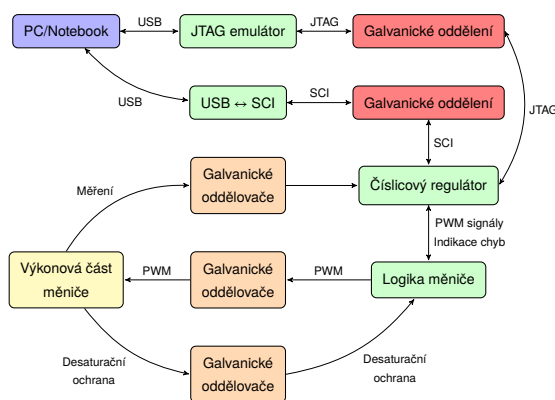
Emulátor JTAGv3 je prvním z prakticky použitelných emulátorů založených na návrhu XDS100 firmy Texas Instruments. Na rozdíl od původního XDS100v1, ze kterého návrh nového emulátoru vychází, neumožňuje tento emulátor využívat funkce boundary-scan (testování funkce/zapojení obvodu bez fyzicky připojených sond, pouze pomocí JTAG sběrnice).



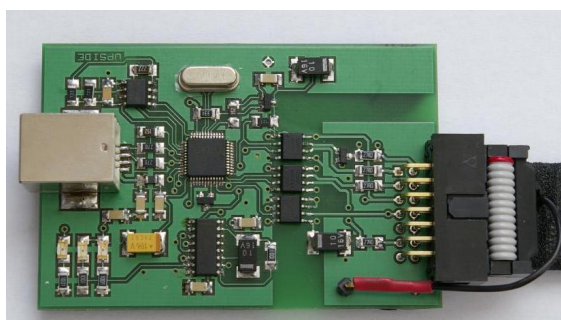
Obr. 3.2.: Vložený USB oddělovač



Obr. 3.3.: Galvanický oddělovač USB sběrnice



Obr. 3.4.: Oddělení na JTAG sběrnici

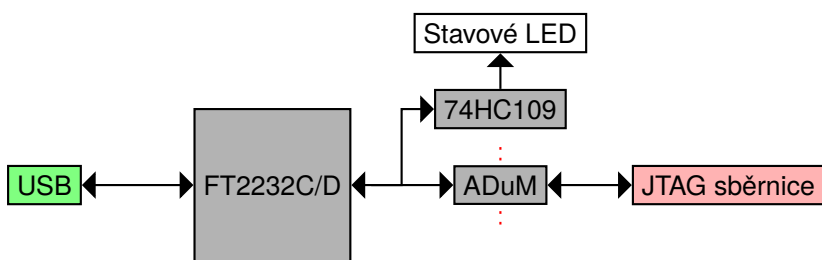


Obr. 3.5.: Emulátor JTAGv3 s oddělenou JTAG sběrnici

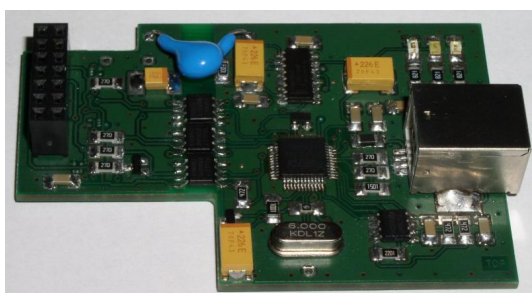
ce), protože nemá hardwarovou podporu pro ovládání pinů EMU0 a EMU1 procesorů řady TMS320F28xxx. Ta byla odstraněna z důvodu snížení výsledné ceny a zjednodušení emulátoru.

Naopak přidává, z pohledu jeho použití v laboratořích výkonové elektroniky, důležité galvanické oddělení JTAG od USB sběrnice, které původní XDS100 poskytuje formou externího velmi nákladného modulu.

Základem emulátoru je obvod firmy FTDI FT2232D, který poskytuje akcelerovaný univerzální synchronní sériový port. Ten podporuje téměř všechny synchronní sériové protokoly (např. JTAG a SPI) s maximální přenosovou rychlostí až 6 Mb/s, podrobné informace viz [47]. Jak bylo zmíněno výše, bylo přidáno galvanické oddělení USB od JTAG sběrnice. K tomuto účelu byly použity obvody ADuM1200BRZ a ADuM1201BRZ [48]. Tyto obvody přidávají průchozí zpoždění signálům JTAG sběrnice, což se projeví na maximální přenosové rychlosti, která klesne z 6 Mb/s na 3,5 Mb/s. Oddělovače zároveň přizpůsobují napěťové úrovně FTDI obvodu a připojeného procesoru, jsou schopny spolehlivě fungovat v rozsahu 2,5 až 5,5 V. Emulátor byl dále doplněn indikací provozního stavu pomocí dvou LED. Blokové schéma viz Obr. 3.6.



Obr. 3.6.: Blokové schéma JTAGv3/4



Obr. 3.7.: JTAG verze 4



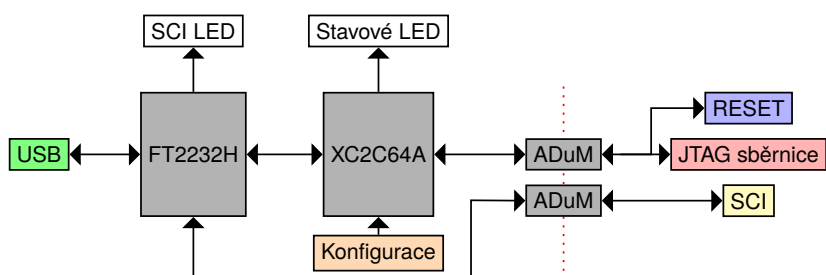
Obr. 3.8.: JTAG verze 5

Z praktických zkušeností při používání bylo zjištěno, že je nutné na propojení JTAG emulátoru s mikroprocesorovým regulátorem použít stíněný kabel. Bez použitého stínění je emulátor schopen spolehlivé funkce až do 200 V ve stejnosměrném obvodu měniče. Pokud byla tato hranice překročena, docházelo k selhání komunikace mezi mikrokontrolérem a vývojovým prostředím. Řešením je obalení plochého propojovacího kabelu stínící folií a spojení tohoto stínění se zemí na straně JTAG emulátoru.

Rozdíly mezi verzí 3 (Obr. 3.5) a 4 (Obr. 3.7) jsou především v odolnosti proti EMI dané provedením plošného spoje, kdy JTAGv4 je primárně určen pro přímé zapojení do JTAG konektoru *eZdsp* kitu (podrobné informace viz [49]) nebo mikrokontrolérového modulu (viz 3.6.1), bez použití propojovacího kabelu, což eliminuje rušení indukované do signálů JTAG sběrnice, také není nutné vyrábět stíněný kabel.

### 3.4. Emulátor JTAGv5

S příchodem nové řady procesorů Hercules TMS570 firmy Texas Instruments a nasazení typu TMS570LS3137 jako hlavního procesoru v systému REMCS, vznikl požadavek na nový JTAG emulátor, který bude podporovat programování a ladění i tohoto nového typu mikroprocesoru. Nová verze emulátoru je založena na designu XDS100v2 [44] firmy Texas Instruments a stejně jako u verzí JTAGv3 a 4 byl návrh vylepšen tak, aby vyhovoval použití v laboratořích výkonové elektroniky. Blokové schéma zobrazuje Obr. 3.9. Emulátor je založen na novější generaci obvodu firmy FTDI FT2232H [50] v kombinaci s doplňkovou lo-



Obr. 3.9.: Blokové schéma JTAGv5

gikou implementovanou v CPLD obvodu Xilinx XC2C64A, JTAG sběrnice je po zpracování v CPLD opět galvanicky oddělena obvody ADuM stejně jako u předchozích verzí. Nabízela se možnost vložit galvanické oddělení již mezi FT2232H a XC2C64, pak by CPLD mohlo fungovat i jako přizpůsobení napěťových úrovní. Od této topologie bylo nakonec ustoupeno, neboť konverzi napěťových úrovní zajistí i samotné ADuM oddělovače v rozsahu 2,5 až 5 V (ADuM120xB). Propojení CPLD - FT2232H by vyžadovalo více oddělovačů, než je potřeba pro oddělení samotné JTAG sběrnice. Výsledné zařízení viz Obr. 3.8.

Emulátor JTAGv5 byl navržen s ohledem na zkušenosti nasbírané při provozu verzí 3 a 4. Použití CPLD Xilinx XC2C64A jako můstku mezi FT2232H a JTAG sběrníci umožnilo implementovat do jeho firmware navíc několik funkcí, které původní XDS100v2 postrádá:

- Indikace aktivity JTAG sběrnice.
- Indikace přítomnosti napájecího napětí v laděné aplikaci.
- Indikace správně zapojeného kabelu.
- Vylepšuje kompatibilitu s jinými softwarovými nástroji, např. UrJTAG (viz sekce 3.4.2), určených pro obecnou práci s JTAG sběrníci.
- Fail-safe funkce - podrobnosti viz 3.4.1.

Indikační funkce slouží uživateli k detekci chyb propojení nadřazené jednotky (PC) s cílovým mikrokontrolérem. Na první pohled je možné rozlišit stav připojeného číslicového regulátoru. Význam jednotlivých stavů indikátorů shrnuje Tab. 3.1.

Jak bylo zmíněno výše, JTAGv5 umožňuje nastavit úroveň kompatibility i s jinými nástroji než je CCS. Především jde o možnost použití emulátoru s programem UrJTAG [51], který lze (mimo jiné) využít jako interpret SVF (serial vector format [52]) souborů. Tyto soubory se dají použít pro programování CPLD nebo FPGA, jako alternativa k binárním formátům programovacích dat.

Emulátor má na desce plošného spoje tři zkratovací propojky, pomocí nichž lze nastavit úroveň kompatibility s originálním designem XDS100v2, význam propojek je pak vysvětlen v Tab. 3.2.

Emulátor JTAGv5 poskytuje galvanicky oddělenou sériovou linku s indikací aktivity RX a TX, která má stejné napěťové úrovně jako JTAG sběrnice, díky tomu ji lze připojit přímo k vývodům mikrokontroléru. Lze tak ušetřit jeden USB port, který by jinak zabíral převodník z USB na SCI. Obě sběrnice, JTAG i SCI jsou schopny fungovat paralelně a navzájem se neovlivňují.

### 3.4.1. Fail-safe funkce emulátoru JTAGv5

Fail-safe funkce byla implementována kvůli situacím, které mohou nastat při provozování měničů na napětích od 200 V výše. Emulátor JTAGv5 je velmi odolný proti EMI generovanému při běžném provozu měničů, ale i tak může dojít k výpadku komunikace mezi vývojovým prostředím (CCS) a mikroprocesorovým regulátorem nebo vývojovým kitem.

Tento chybový stav lze detekovat přímo v JTAG emulátoru tak, že je periodicky kontrolována přítomnost impulzů minimálně na jednom ze signálů JTAG sběrnice. Debugger, který je součástí CCS, totiž periodicky zjišťuje stav mikrokontroléru i při zastaveném programu. V případě JTAGv5 je možné do CPLD implementovat watchdog, který je resetován s každým impulzem na signálu TCK JTAG sběrnice. Jakmile dojde k přetečení čítače watchdogu (což znamená absenci pulzů na TCK signálu), mikrokontrolér je uveden do stavu reset. Hlavní nevýhoda tohoto principu spočívá v závislosti prodlevy této ochrany na četnosti dotazů debugeru na stav laděného zařízení. V případě debugeru CCS je tato latence maximálně 10 s při zastaveném programu. Z důvodu bezpečnostní rezervy je prodleva watchdogu nastavena na 12 s. Takovýto čas se jeví jako nepraktické, zde se však nejedná o ochranu reagující na chybový stav měniče, ale o zajištění korektního vypnutí řízeného hardwaru bez zásahu obsluhy, když nad ním ztratí možnost kontroly z nadřazené aplikace (zde debugger CCS). Pokud nastane chybový stav, je mikroprocesor držen ve stavu reset do doby, než chybový stav odezní.

Čistě hardwarovou bezpečnostní funkcí je, že odpojení USB kabelu nebo ztráta napájení na USB straně emulátoru opět mikrokontrolér uvede do stavu reset a v něm pak zůstává do doby, než je napájení znovu obnoveno.

Tab. 3.1.: Význam stavových indikátorů emulátoru JTAGv5

Barva stavové LED	Nesvíí	Bliká	Svíí trvale
Zelená	Cílová aplikace nemá napájení	Cílová aplikace má napájení, ale není inicializována	Cílová aplikace je připravena
Modrá	JTAG sběrnice neaktivní	JTAG sběrnice přenáší data	JTAG sběrnice přenáší data
Žlutá	Reset není aktivní	—	Reset je aktivní

Tab. 3.2.: Funkce konfiguračních propojek JTAGv5

Konektor H7	Funkce	Výchozí stav (nezkratováno)
SW1 (1-2)	Povoluje funkci LOOPBACK pro testování funkce emulátoru	LOOPBACK trvale vypnutý
SW2 (3-4)	Povoluje klopný obvod pro detekci napájení cílového zařízení	Vypnuto, stav napájení a připojení či nepřipojení kabelu k cílového zařízení se projeví okamžitě
SW3 (5-6)	Vypíná funkci fail-safe	Zapnuto, funkce fail-safe je aktivní. Pokud je zapojen resetovací obvod, tak je mikrokontrolér držen ve stavu reset v případech zmiňovaných v podkapitole 3.4.1

Aby bylo možno funkci fail-safe využívat, musí být propojen signál  $\overline{\text{RESET}}$  procesoru s emulátorem, u MLC interface (od DPS revize 0.2 a výše) je příslušný konektor součástí DPS.

### 3.4.2. Nahrávání programovatelných logických obvodů pomocí JTAGv3/4/5

Pro zvýšení univerzality vyvinutého JTAG emulátoru byly prozkoumány možnosti nahrávání firmware do CPLD a FPGA. Na základě této analýzy byla navržena aplikace umožňující programování těchto obvodů.

Jako základní jádro vyvinuté programovací aplikace je použit nástroj třetí strany: UrJTAG [51]. Ten má zabudovány základní podporu pro JTAG emulátory založené na referenčním designu s obvody FT2232C/D/H a zároveň má zabudovaný i interpret SVF. Tím odpadla nutnost návrhu složitějšího parseru pro SVF soubory a také nízkoúrovňových funkcí pro obsluhu JTAG emulátorů. Úpravou zdrojových kódů aplikace UrJTAG (je šířena pod svobodnou licenci) byla přidána přímá podpora výše zmiňovaných emulátorů JTAGv3/4/5, včetně korektního reportování chybových stavů (např. absence napájení cílového obvodu a přerušené JTAG sběrnice). Takto upravený program pak slouží jako exekutivní jádro vyvinuté grafické aplikace *PLD loader*, která jej obaluje a nahrazuje uživatelsky náročnější příkazovou řádku (Obr. 3.10) aplikací UrJTAG grafickým rozhraním dle Obr. 3.11. Vyvinutý program je multiplatformní, podporované platformy jsou Linux a Windows.

Vyvinutá aplikace *PLD loader* používá jako primární vstupní formát souborů typ SVF, který je možné přímo předat exekutivnímu jádru. Pokud však má uživatel nainstalovány i nástroje od firem Altera nebo Xilinx, pak je možné do programovatelných logických obvodů nahrávat i binární formy firmwaru. *PLD loader* zajistí jejich konverzi pomocí nástrojů výrobce programovatelného obvodu na formát SVF a ten pak předá exekutivnímu jádru UrJTAG, který data nahraje do příslušného obvodu. Situaci naznačuje Obr. 3.12. Druhotná funkce spočívá



```

tmp : mc - Konsole
Soubor Úpravy Pohled Záložky Nastavení Nápověda
zmatkar@zmatkar-nb2:/tmp$ jtag

UrJTAG 0.10 #2039
Copyright (C) 2002, 2003 ETC s.r.o.
Copyright (C) 2007, 2008, 2009 Kolja Waschk and the respective authors

UrJTAG is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
There is absolutely no warranty for UrJTAG.

warning: UrJTAG may damage your hardware!
Type "quit" to exit, "help" for help.

jtag> cable JTAGv5
Connected to libftdi driver.
JTAGv5: JTAG Mode Initialization OK!
jtag> detect
IR length: 10
Chain length: 1
Device Id: 00000010000011110100000011011101 (0x020F40DD)
  Filename: /home/zmatkar/.jtag/bsdl/EP3C40Q240.BSD
jtag> svf mlc_interface.svf progress
warning: unimplemented mode 'ABSENT' for TRST
detail: Parsing 9430/9432 ( 99%)detail:
detail: Scanned device output matched expected TDO values.
jtag>
  
```

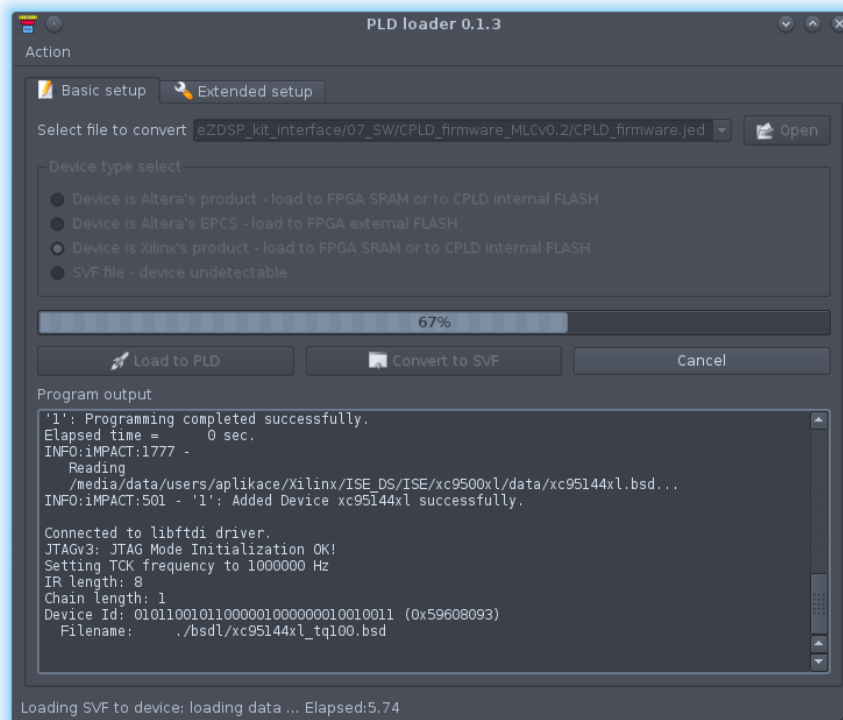
Obr. 3.10.: Konzolové rozhraní programu UrJTAG s přidanou podporou pro emulátor JTAGv5

Tab. 3.3.: Přehled nahrávacích časů pro PLD na MLC interface

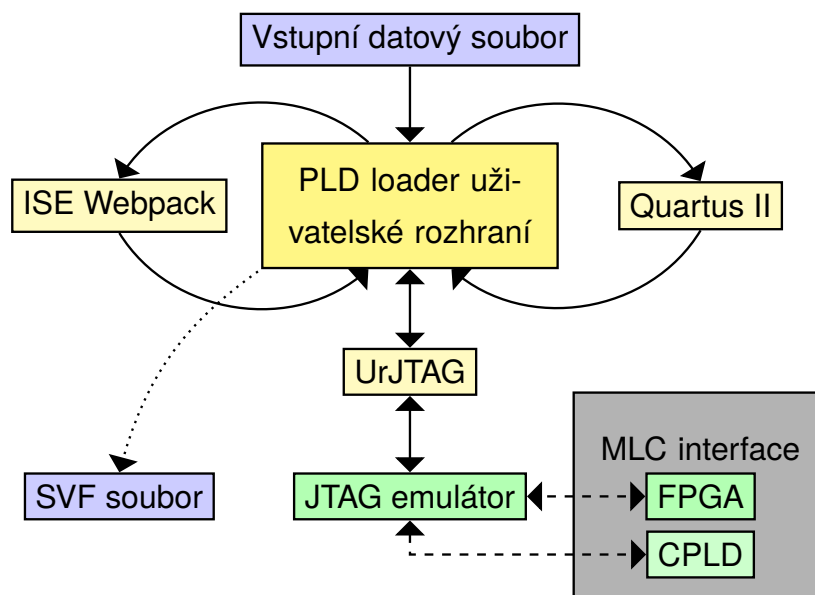
Obvod	Čas nahrání
XC95144XL	6 s
EP3C40 (do SRAM)	2 s
EP3C10 (do SRAM)	2 s
EPCS16 (přes SFL entitu v EP3C40)	250 s
EPCS16 (přes SFL entitu v EP3C10)	190 s

v možnosti provést pouze konverzi z binární formy dat pro PLD do SVF formátu. Tato funkce je využita v případě, že je k dispozici nová verze firmware pro systémové CPLD a je tak umožněno uživateli mikroprocesorového regulátoru MLC interface si tento firmware přehrát pouze pomocí PLD loaderu bez nutnosti mít nainstalované nástroje třetích stran jako jsou Quartus a ISE WebPack.

Vyvinutá aplikace byla otestována s programovatelnými logickými obvody přítomnými na MLC interface a dosahované rychlosti nahrávání jsou vyhovující, s výjimkou programování externí FLASH paměti FPGA (viz Tab 3.3). Toto znevýhodnění se však prakticky neprojevuje, neboť pro ladění designu v FPGA se využije především nahrávání do vnitřní paměti a poté se finální verze designu nahraje do konfigurační FLASH paměti. Aplikace PLD loader je optimalizovaná pro přímočaré použití s mikroprocesorovým regulátorem MLC interface a nevyužívá všechny možnosti poskytované aplikací UrJTAG, jako například skenování logických úrovní na jednotlivých pinech programovatelného logického obvodu.



Obr. 3.11.: Vyvinutá aplikace PLD loader pro nahrávání programovatelných logických obvodů



Obr. 3.12.: Struktura programu PLD loader a jeho napojení na hardware

### 3.5. Multi Level Converter Interface - mikroprocesorový regulátor pro víceúrovňové měniče a výpočetně náročné algoritmy regulace v pohonech

Multi Level Converter interface je vývojový prostředek speciálně navržený zejména pro aplikace s víceúrovňovými měniči, kde je potřebný velký počet synchronních PWM výstupů, a dále na aplikace s vysokými výpočetními nároky v elektrických pohonech, jako jsou zejména matematické modely, algoritmy identifikace parametrů motoru (estimátory), atd.

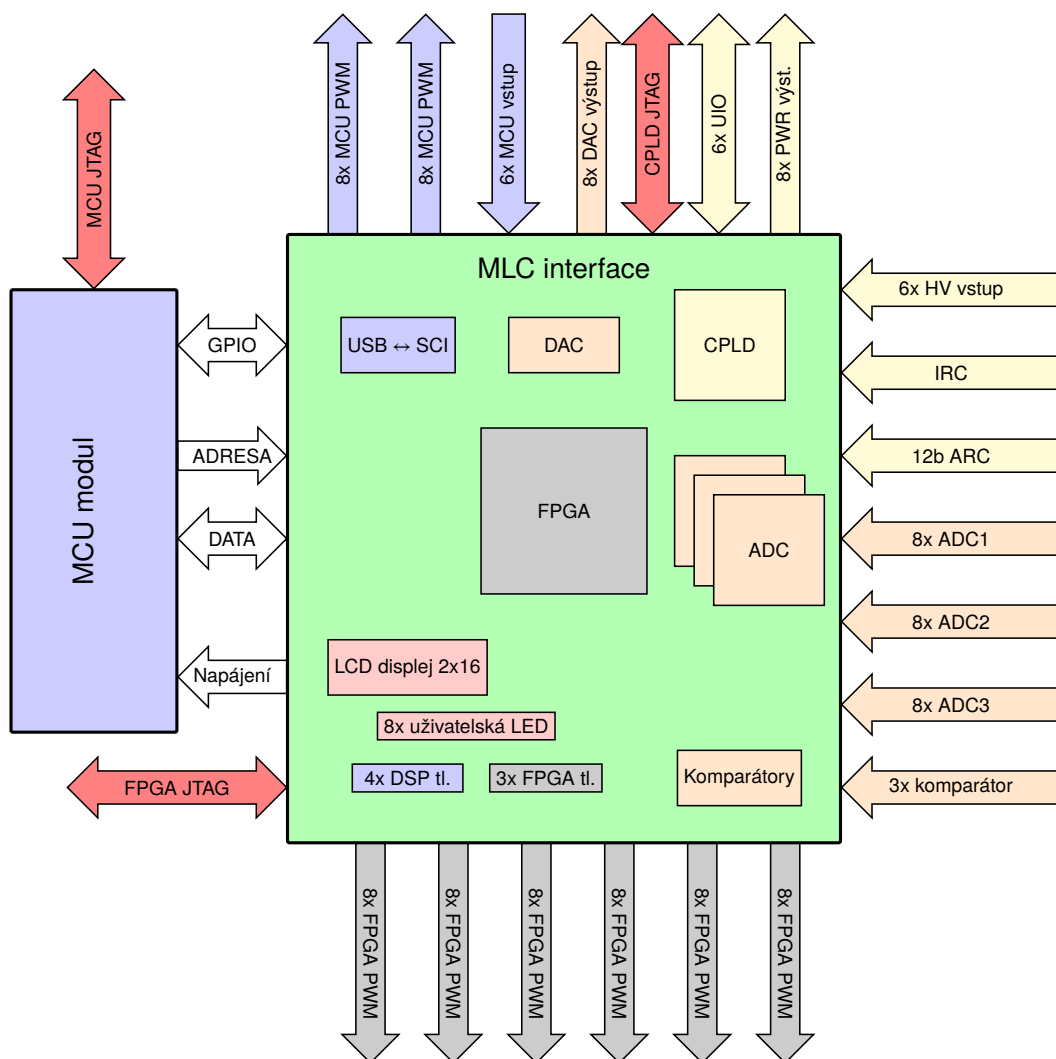
#### 3.5.1. Topologie MLC interface

Obr. 3.13 ukazuje blokové schéma mikroprocesorového regulátoru MLC interface. Ten se skládá ze dvou desek plošných spojů (DPS), jeden modul s vybraným mikrokontrolérem (viz 3.6.1) a základní deska MLC interface. Z hlavních komponent základní desky je nutné zmínit zejména aplikační FPGA typu EP3C40 firmy Altera [53], systémové CPLD XC95144XL od výrobce Xilinx [54], osmikanálový digitálně-analogový převodník AD5328 firmy Analog Devices [55], tři osmivstupé analogově-digitální převodníky s plně diferenciálními vstupy AD7609 [56], analogové komparátory pro detekci průchodu signálu nulou, LCD displej 2x16 znaků, uživatelská tlačítka a obvody pro přizpůsobení napěťových úrovní vstupů a výstupů. Postavený MLC interface viz Obr. A.1.

Koncepce platformy je založena na propojení FPGA, CPLD a ADC pomocí paralelní 16 bitů široké datové a 9-ti bitové adresové sběrnice. Pro lepší kompatibilitu s předchozím vývojovou platformou primárně určenou pro TMS320F2812 (viz práce [36]) byl MLC interface navržen tak, že není nutné mít naprogramovaný firmware v FPGA pro základní funkčnost hardwaru na základní desce MLC interface. Logika MLC interface je rozdělena na dvě nezávislé části: aplikační zastoupenou FPGA a systémovou v podobě CPLD. Každý z obvodů má svou vymezenou část adresního prostoru, CPLD má konkrétně přiřazeny adresy 0x0000 až 0x000F, uživatel FPGA má k dispozici adresy 0x0010 až 0x01FF. Tyto adresy jsou z hlediska mikroprocesoru mapovány do paměťového prostoru určeného pro I/O zařízení a jedná se tedy o offsety od základní adresy dané části paměti dle Tab. 3.4. Všechny vybrané paměťové zóny mikrokontrolérů se vyznačují tím, že mají zaručené pořadí operací čtení a zápisu.

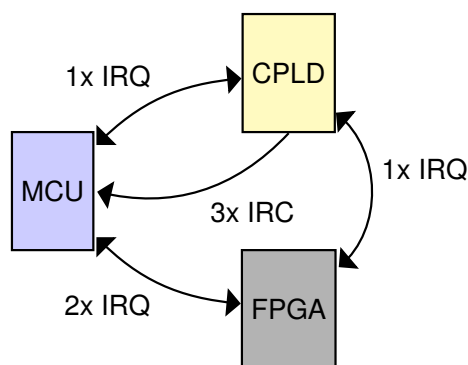
Tab. 3.4.: Základní adresy pro přístup k externím periferiím

Typ mikrokontroléru	Základní adresa
TMS320F2812	0x00004000
TMS320F28335	0x00004000
TMS570LS3137	0x60000000



Obr. 3.13.: Blokové schéma MLC interface

Mimo adresní a datovou sběrnici jsou jednotlivé obvody propojeny přímými signály (Obr. 3.14), např. pro vyvolání přerušování, synchronizaci běhu nebo předávání hodinového signálu. Mezi mikrokontrolérem a FPGA jsou tyto signály dva, mikrokontrolér a CPLD mají k dispozici jeden signál a FPGA je s CPLD spojeno jedním přímým signálem. Mimo adresovou sběrnici jsou vedeny také zpracované signály z IRC čidla, které jsou napojeny přímo na vstupy QEP jednotky procesoru. MLC interface používá jeden globální signál s pevně danou funkcí globálního povolení  $\overline{GLOBAL\_ENABLE}$ . Tento signál je použit pro blokování výstupů CPLD a FPGA po resetu mikrokontroléru. Využívá toho, že mikrokontrolér má po resetu své vývody ve třetím stavu, přizvedávacím odporem je signál nastaven na logickou 1 a tím indikuje oběma PLD obvodům, že mají své výstupy přepnout do neaktivního stavu. Jakmile uživatel spustí kód v procesoru, tak pomocí funkce `MLC_global_enable()` (tu poskytuje ovladač k MLC interface, viz 3.5.4) mikrokontrolér signalizuje PLD obvodům, že mají své výstupy aktivovat. Výhodou tohoto přístupu je téměř okamžitá reakce na reset mikrokontrolé-



Obr. 3.14.: Propojení řídicích obvodů přímými signály

ru bez ohledu na jeho zdroj. Stejným způsobem se projeví reset hardwarový nebo vyvolaný uživatelem pomocí JTAG emulátoru (restart programu z CCS).

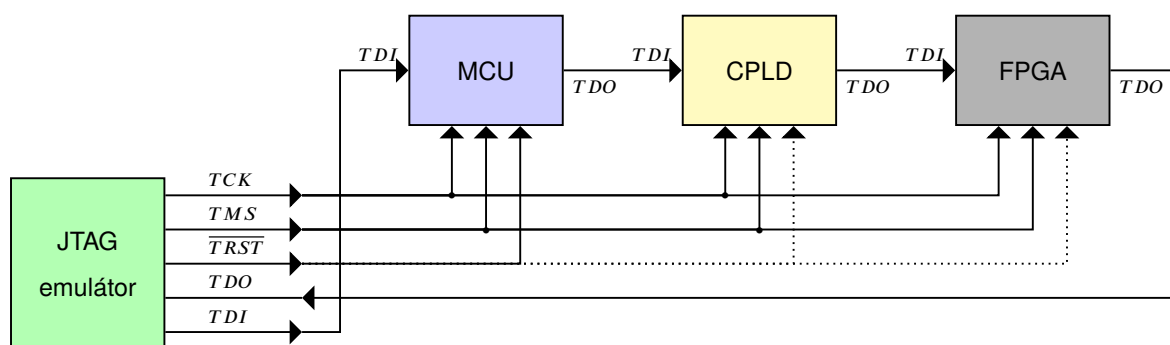
### 3.5.1.1. Vstupy a výstupy dostupné na MLC interface

MLC interface poskytuje uživateli šest univerzálních vstupů připojených přes vstupní oddělovač typu CD4050 přímo na vývody mikrokontroléru. Tyto vstupy mají odolnost až do 30 V. Z mikrokontroléru jsou přes přizpůsobovače úrovní CD4504B vyvedeny dvě skupiny PWM výstupů. Napěťová úroveň těchto výstupů je nastavitelná na 3,3 V/5 V/15 V. Na každý z PWM výstupů je přes posilovač připojena LED dioda, díky kterým je možné vizuálně kontrolovat funkci jednotlivých PWM výstupů. Na každou osmici PWM výstupů připadá jeden vstupní  $\overline{\text{FAULT}}$  signál indikující chybu měniče.  $\overline{\text{FAULT}}$  vstupy mají stejně jako univerzální vstupy odolnost do napětí 30 V.

Přímo na vývody SCI (serial communication interface) modulu mikrokontroléru je zapojen i převodník USB  $\leftrightarrow$  SCI založený na výrobcem doporučeném zapojení obvodu FT232RL [57]. Linky TX a RX jsou galvanicky odděleny od USB sběrnice obvodem ADuM1201B, což v kombinaci s odděleným JTAG emulátorem kompletně izoluje nadřazenou jednotku (PC/notebook) od samotného MLC interface i řízené měničové sestavy. RX linka je odbočena do FPGA a TX linka může být pomocí zkratovací propojky odpojována od mikrokontroléru a zavedena do FPGA.

Digitálně-analogový převodník umožňuje výpis hodnot proměnných z mikrokontroléru nebo FPGA na osciloskopu v reálném čase. Digitálně-analogový převodník je připojen přes SPI sběrnici a multiplexer, který jej umožňuje připojit k FPGA místo mikrokontroléru. Převodník poskytuje uživateli osm nezávislých výstupů s rozsahem 0 až 5 V s rozlišením 12 bitů.

Uživatel má dostupný rozšiřující konektor pro jednoduché připojení dodatečných modulů do topologie MLC interface. Na tento konektor je vyvedena datová i adresová sběrnice, ovládací signály těchto sběrnic ( $\overline{\text{WR}}$ ,  $\overline{\text{RD}}$  atd.) a tři  $\overline{\text{CS}}$  signály pro externí moduly, které poskytuje firmware v CPLD. Připojený modul tedy nemusí implementovat vlastní dekodér ad-



Obr. 3.15.: Příklad připojení obvodů na jednu JTAG sběrnici

resy, což zjednoduší jeho obvodový návrh. Konektor poskytuje i základní napěťové hladiny dostupné na MLC interface: +3,3 V, +5 V, -5 V.

### 3.5.1.2. Zapojení JTAG sběrnice

JTAG sběrnici [58] je možné zapojit do tzv. "JTAG chainu", kdy výstup *TDO* z jednoho obvodu zapojíme do vstupu *TDI* dalšího integrovaného obvodu (Obr. 3.15). Tímto způsobem je možné zapojit na jednu JTAG sběrnici několik obvodů. Spojení všech programovatelných součástí je výhodné v případě, že je požadováno jedním JTAG emulátorem programovat všechny zřetěžené obvody. Sériovým spojením několika součástí však narůstá bitová délka řetězce, což v důsledku vede na redukci komunikační rychlosti po JTAG sběrnici. Sběrnice má zároveň omezení na svou délku, od 15 cm je doporučeno použít posilovače signálů.

Řešení u MLC interface poskytuje pro každý ze dvou programovatelných obvodů (FPGA a CPLD) separátní JTAG konektor a na procesorovém modulu je umístěn další (viz 3.13). Separátní JTAG sběrnice jsou krátké a odolnější vůči rušení vznikající při provozu měničů než jeden dlouhý JTAG řetězec. Díky tomuto oddělení JTAG sběrnic je možné simultánně nahrávat a ladit softwarové vybavení v FPGA i v mikrokontroléru.

### 3.5.2. Firmware systémového CPLD a jeho vstupy a výstupy

Systémové CPLD zajišťuje nízkoúrovňové funkce nutné pro základní použití MLC interface. Poskytuje uživateli sadu registrů a vstupů, případně výstupů, mapovaných do paměti mikrokontroléru, pomocí nichž lze ovládat analogově-digitální převodníky, číst hodnoty z absolutního čidla polohy a ostatních vstupů a ovládat uživatelské výstupy. Blokové schéma firmwaru implementovaného do CPLD je zobrazeno na Obr. 3.18.

Základem firmware je dekodér adres, ten zajišťuje multiplexaci vnitřních dat z/na externí datovou sběrnici sdílenou s mikrokontrolérem. Tabulky Tab. 3.5, 3.6, 3.7 shrnují adresy implementované ve firmwaru a v ovladači pro MLC interface, zápis či čtení z nedefinovaných adres nemá definovaný výsledek. Z tabulky je patrné, že se některé adresy pro čtení a zápis shodují, i když mají pokaždé jiný význam. Např. zápis na adresu 0x0000 nastavuje AD

Tab. 3.5.: Adresy registrů pro zápis implementovaných v CPLD

Název	Offset adresy	Funkce
WRITE_AD_CONF	0x0000	Konfigurační registr AD převodníků
WRITE_PWR_OUT	0x0001	Ovládací registr výkonových výstupů
WRITE_SETRES_AD	0x0002	Přepne ADC do stavu reset
WRITE_CLRRES_AD	0x0003	Přepne ADC do stavu funkčního stavu
WRITE_DBGLEDS	0x0004	Ovládací registr DEBUG výstupů

Tab. 3.6.: Adresy registrů pro čtení implementovaných v CPLD

Název	Offset adresy	Funkce
READ_VER	0x0000	Registr verze firmware v CPLD
READ_AD1	0x0001	Adresa pro čtení AD CH1
READ_AD2	0x0002	Adresa pro čtení AD CH2
READ_AD3	0x0003	Adresa pro čtení AD CH3
READ_HV_INPUT	0x0004	Adresa pro čtení HV vstupů
READ_ARC	0x0005	Adresa pro čtení ARC vstupu
READ_PCB_REV	0x000F	Registr s verzí DPS

převodníky, ale čtení ze stejné adresy vrací verzi firmwaru nahraného v CPLD. Toto chování je implementováno záměrně, aby byla redukována velikost firmware v CPLD, které má mnohem omezenější prostředky než FPGA. K rozlišení funkce registru na jednotlivých adresách slouží konstanty definované v rámci ovladače pro MLC interface (viz 3.5.4). V ovladači *MLC\_interface\_lib* se tyto adresy rozlišují pomocí prefixu: *WRITE\_* označuje adresy jen pro zápis, *READ\_* jen pro čtení, *RW\_* mají implementované čtení i zápis pro funkčnost naznačenou zbylou částí názvu.

Primární funkcí firmware v CPLD je umožnit obsluhu a nastavení AD převodníků. Dekodér adres umožňuje adresovat jeden ze tří AD převodníků AD7609 [56] a číst z něj naměřené hodnoty. V okamžiku nastavení příslušné adresy mikrokontrolérem (dle tabulky Tab. 3.6) se vybaví  $\overline{CS}$  vstup jednoho ze tří převodníků a lze z něj číst data. Analogově-digitální převodník AD7609 nedisponuje vnitřním konfiguračním registrem, konfigurace se volí logickými úrovněmi na příslušných vstupech. Nastavování je zajištěno opět pomocí CPLD, zápisem do registru na adrese *WRITE\_AD\_CONF* lze nastavit vstupní rozsah převodníku, převzorkování (to provádí analogově-digitální převodník sám) a způsob spouštění převodu.

Dva důležité registry *READ\_VER* a *READ\_PCB\_REV* poskytují základní informace o verzi firmware v CPLD a variantě DPS. Uživatel tak může softwarově vybavení přizpůsobit možnostem hardwaru. Umístěním těchto hodnot v CPLD je zaručeno, že budou vždy odpovídat dané implementaci hardware i systémového firmware.

Tab. 3.7.: Adresy registrů pro čtení i zápis implementovaných v CPLD

Název	Offset adresy	Funkce
RW_CS_EXT1	0x0006	Čtení/zápis externího modulu 1
RW_CS_EXT2	0x0007	Čtení/zápis externího modulu 2
RW_CS_EXT3	0x0008	Čtení/zápis externího modulu 3
RW_UIO	0x0009	Čtení/zápis UIO pinů
RW_UIO_CONF	0x000A	Čtení/zápis konfigurace UIO pinů

### 3.5.3. Základní firmware pro FPGA

Pro MLC interface byl vyvinut základní firmware pro FPGA. Ten usnadňuje uživateli vývoj specializovaných entit bez nutnosti podrobně znát hardware MLC interface (např. funkce konkrétních pinů FPGA). Jeho blokové schéma je na Obr. 3.19, firmware implementuje stejně jako varianta pro systémové CPLD:

- Dekodér adres z externí paměťové a datové sběrnice.
- Obsluhu signálu reset a  $\overline{\text{GLOBAL\_ENABLE}}$ .

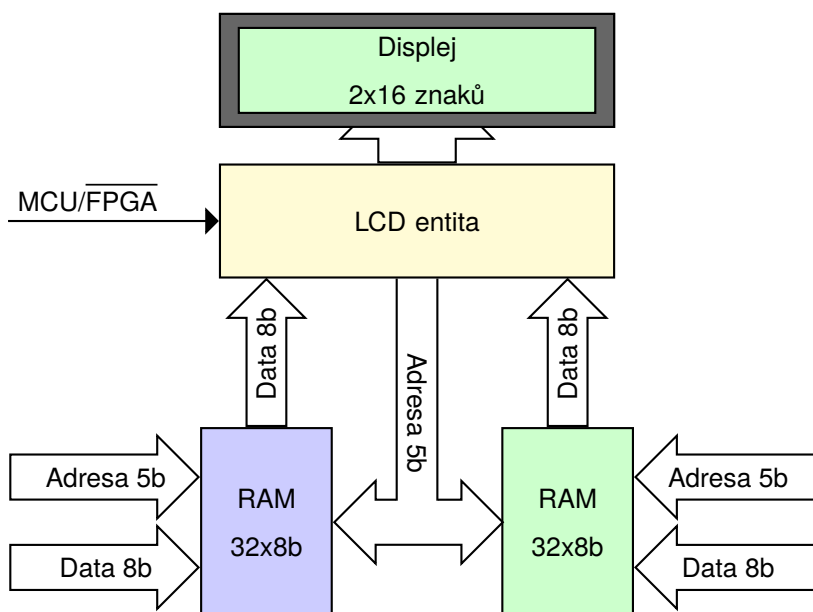
Navíc jsou přítomny entity přidávající podporu pro periferie připojené k FPGA:

- Asynchronní bufferovanou entitu pro obsluhu znakového LCD displeje.
- Entitu pro snadné přepínání signálů zobrazovaných pomocí uživatelských LED.
- Generátory hodin pro vnitřní entity (100/10/1/0,1 MHz), návrh FPGA má dvě hodinové domény, jednu synchronní s adresovou a datovou sběrnici o frekvenci 37,5 MHz (a tedy s vnitřními hodinami mikrokontroléru) a jednu asynchronní 10 MHz.
- Entitu Serial Flash Loader, která umožňuje nahrávání firmware do externí datové FLASH paměti přes JTAG sběrnici.

Přehled aktuálně obsazených adres FPGA ukazuje Tab. 3.8.

První jmenovaná entita zajišťuje obsluhu LCD displeje 2x16 znaků. Funkčně se jedná o RAM s velikostí 32 bytů, která má jednu bránu pro zápis a druhou čtení (viz Obr. 3.16). Paměti jsou v entitě dvě, kdy jedna je napojena na dekodér adres a zapisuje do ní mikrokontrolér přes externí sběrnici, druhý blok paměti pak mohou využít entity v FPGA pro výpis na displej. Přepínání se provádí signálem  $\overline{\text{MCU/FPGA}}$  entity LCD displeje. Obsah paměti vybrané tímto signálem se cyklicky vypisuje na LCD displej. Protože je vždy vypsáno všech 32 znaků, není třeba obsah displeje před zápisem mazat, což zabraňuje problikávání zobrazovaných údajů. Dvoubránové RAM slouží jako vyrovnávací paměti mezi rychlou externí datovou sběrnici mikrokontroléru (cyklus 100 ns), resp. uživatelskými entitami v FPGA, a pomalu komunikujícím řadičem znakového LCD displeje (cyklus 5  $\mu\text{s}$ ).





Obr. 3.16.: Blokové schéma entity pro obsluhu LCD displeje

Tab. 3.8.: Adresy registrů implementovaných v základním designu FPGA

Název registru	Adresa	Funkce
READ_FPGA_VER	0x0010	registr verze firmwaru v FPGA
RW_FPGA_LED_REG	0x0011	registr nastavení LED entity
WRITE_LCD_DISP	0x0012	registr pro výpis na LCD displej

Pro výpis na displej z mikrokontroléru je s výhodou využita 16 bitů široká datová sběrnice MLC interface. Spodní byte definuje znak, který je třeba zobrazit a horní byte určuje na jakou pozici displeje bude znak vypsán (Obr. 3.17). Znaky se indexují lineárně od 0 do 31 v případě pro použitý displej 2x16 znaků (nejsou tedy rozlišeny řádky). Takto definovaný komunikační formát umožňuje entitě obsloužit displej až o 256 znacích.



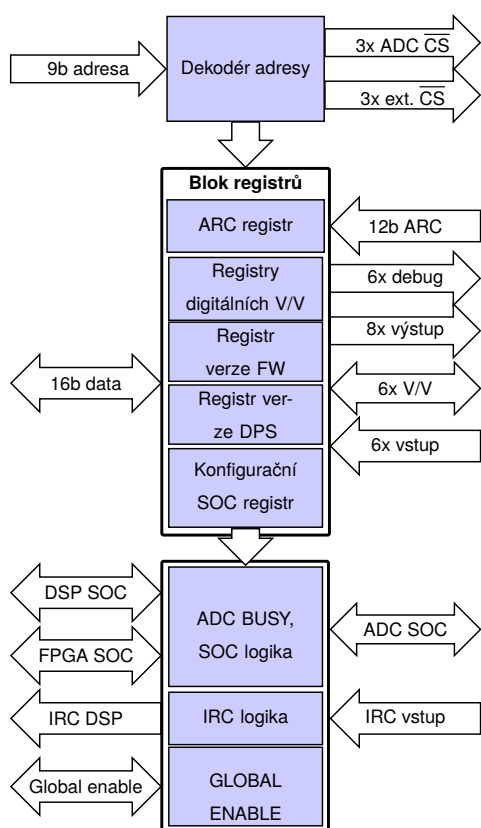
Obr. 3.17.: Struktura dat posílaných z mikrokontroléru do entity LCD displeje

LED entita umožňuje uživateli kontrolovat osm LED připojených k FPGA. Registr *RW\_FPGA\_LED\_REG* umožňuje nastavit chování této entity, potažmo LED k ní připojených. Nyní je možné LED ovládat přímo zápisem do spodního byte registru nebo nastavením horního byte LED napojit přímo na výstupy jedné z šesti PWM bran ovládaných pomocí FPGA.

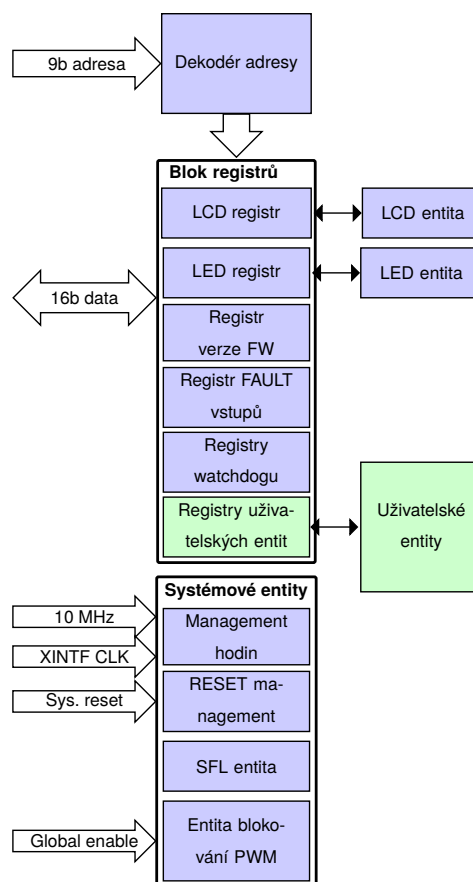
FPGA design má v základu i několik systémových entit nedostupných přes datovou sběrnici. Ty zajišťují dělení a násobení základních hodin (synchronních s mikrokontrolérem či asynchronních 10 MHz), synchronizaci externího resetu. Entita SFL loader umožňuje nahrávání sériové FLASH paměti, ve které se uchovává firmware pro FPGA. Pokud uživatel

používá nástroje výrobce Altera (ByteBlaster + Quartus programmer), pak není tato entita nutná, avšak při použití programu PLD loader je její přítomnost v FPGA klíčová pro úspěšné nahrání firmware do obvodu.

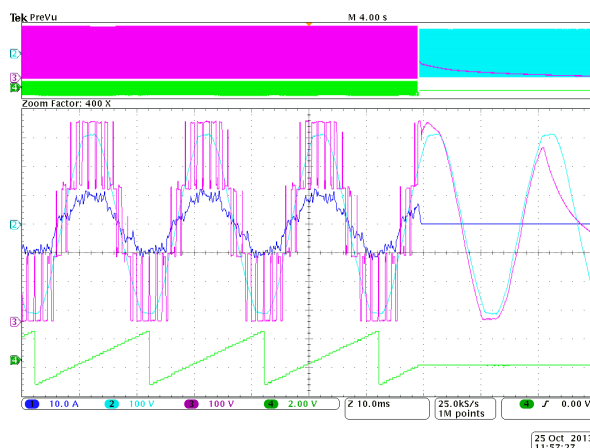
Poslední systémovou entitou je blokování PWM výstupů z FPGA. V případě, že není signál `GLOBAL_ENABLE` v log.0, jsou výstupy v neaktivním stavu (log.0). Toto chování je klíčové pro korektní provozování měničů, pokud by se PWM výstupy neblokovaly v okamžiku, kdy je chování mikrokontroléru nedefinované, FPGA by dále generovalo výstupní signály, ovšem vektor výstupního napětí by se zastavil, což by vedlo k nadproudu ve výkonové části měniče. Reakce FPGA a potažmo měniče na softwarový reset mikrokontroléru je zobrazena na Obr. 3.20. Zde je vidět, že v okamžiku resetu mikrokontroléru se okamžitě vypnou výkonové prvky a proud tekoucí obvodem plynule klesne k nule. Pro měniče proudového typu je nutné toto chování upravit tak, aby se naopak všechny prvky sepnuly. Podobnou funkci jako signál `GLOBAL_ENABLE` zastává i watchdog implementovaný v FPGA, který po uplynutí uživatelem zvolené doby uvede PWM výstupy do bezpečného stavu. Nevýhoda watchdogu spočívá v dodatečném vytížení mikrokontroléru, který musí zajišťovat jeho pravidelné resetování pomocí zápisu na externí sběrnici. Watchdog má minimální časovací krok  $103 \mu s$ ,



Obr. 3.18.: Blokové schéma CPLD obvodu na MLC interface



Obr. 3.19.: Blokové schéma FPGA obvodu na MLC interface



Obr. 3.20.: Ukázka korektního vypnutí NPU s topologií 4L-FLC při softwarovém resetu mikrokontroléru, CH1 (modrá): vstupní fázový proud usměrňovače [10 A/d], CH2 (světle modrá): vstupní fázové napětí usměrňovače [100 V/d], CH3 (fialová): výstupní fázové napětí usměrňovače [100 V/d], CH4 (zelená): řídicí úhel  $\gamma$  [38°el./d]

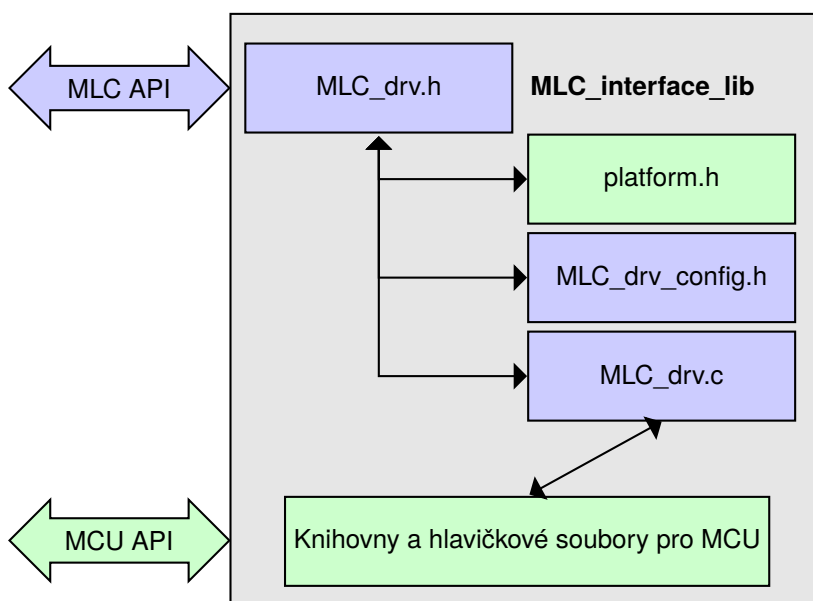
časovací prodleva je nastavitelná uživatelem v  $2^{16}$  krocích, a maximální dosažitelný čas je pak 6,75 s.

Navržený firmware (resp. celý projekt pro vývojové prostředí Quartus) slouží jako základní stavební kámen pro případné rozšíření funkcionality FPGA uživatelem. Pak není nutné řešit nízkoúrovňové hardwarově závislé části firmware, jako je mapování pinů, komunikace po datové sběrnici s mikrokontrolérem, a může se zaměřit pouze na implementaci svých funkčních bloků. Ty pak pouze namapuje do adresního prostoru mikrokontroléru přidáním zvolených adres do procesů čtení a zápisu datové sběrnice hlavní entity.

#### 3.5.4. Programové vybavení pro mikrokontrolér - `MLC_interface_lib`

Používání MLC interface je zjednodušeno pomocí knihovny ovladačů `MLC_interface_lib`. Jedná se o multiplatformní sadu maker a funkcí napsaných v jazyce C, které obsluhují hardware přítomný na MLC interface tak, aby uživatel nemusel mít podrobné znalosti zapojení periférií MLC interface. Knihovna je dostupná pro všechny tři platformy mikrokontrolérových modulů.

`MLC_interface_lib` poskytuje dvě aplikační programová rozhraní (API). První `MCU API`, dodávané výrobcem, obsahující pojmenování registrů mikrokontroléru, mapu paměti a základní nízkoúrovňové funkce úzce vázané na architekturu mikrokontroléru, které se liší dle použitého obvodu a druhé rozhraní `MLC API`, které je naopak zcela nezávislé na použité platformě mikrokontroléru. Zvolená struktura pak umožňuje sdílet zdrojové soubory definující `MLC API` pro všechny podporované platformy a tím lze udržet konzistentní API mezi platformami, za cenu použití podmíněného překladače. Struktura ovladače je znázorněna na Obr. 3.21.



Obr. 3.21.: Struktura knihovny MLC\_interface\_lib

Definice *MLC API* a základních adres využitých ve firmwarech pro CPLD a FPGA je obsažena v souboru *MLC\_drv.h*, implementaci obsahuje soubor *MLC\_drv.c*. Pomocí makra *MCU\_TYPE* v souboru *platform.h* je definován typ procesoru. Makro je dále použito v implementační části pro podmíněný překlad platformně závislých částí kódu knihovny. Konfigurační volby knihovny lze najít v souboru *MLC\_drv\_config.h*. Dokumentace je součástí hlavičkového souboru *MLC\_drv.h*, je použit program Doxygen [59], který ze speciálně formátovaných komentářů jazyka C umí generovat dokumentaci ve formátu HTML či  $\LaTeX$  (který lze dále převést do formátu PDF).

### 3.5.5. Aplikační rozhraní MLC API

Aplikační rozhraní je rozsáhlá knihovna funkcí, která abstrahuje hardware MLC interface. Kompletní dokumentace rozhraní je velmi rozsáhlá, proto zde bude uveden pouze stručný popis základních funkcí tohoto API. Plná verze dokumentace a zdrojové kódy pro všechny podporované platformy jsou dostupné na [60].

#### 3.5.5.1. Základní makra a struktury MLC API

Základními makry a funkcemi poskytovanými aplikačním rozhraním jsou *MLC\_WRITE(...)* a *MLC\_READ(...)*, resp. *MLC\_write(...)* a *MLC\_read(...)* (viz Výp.3.1). Funkčnost maker i funkcí je ekvivalentní.

Tato dvě makra, resp. funkce, zajišťují unifikovaný přístup k externí sběrnici, nezávislý na použitém procesoru resp. na jeho adrese externích periférií (viz Tab.3.4). Obecně lze k registrům entit v FPGA či CPLD přistupovat i přímým zápisem na danou adresu, avšak

```

#define MLC_WRITE(addr, data) *((volatile uint16_t*)BASE_ADDRESS+addr)=(uint16_t)
    (data)
#define MLC_READ(addr) *((volatile uint16_t*)BASE_ADDRESS+addr)
__inline void MLC_write(uint16_t addr, uint16_t data) {*((volatile uint16_t*) (
    BASE_ADDRESS+addr))=(uint16_t) data;}
__inline uint16_t MLC_read(uint16_t addr) {return *((volatile uint16_t*) (
    BASE_ADDRESS+addr));}

typedef struct{
    uint16_t cpld_fw_ver;
    uint16_t fpga_fw_ver;
    uint16_t pcb_hw_ver;
    uint16_t adc_conf;
    uint16_t mod_fw_ver;
} mlc_info_t;

```

Výp. 3.1: Základní makra/funkce a struktury ovladače

uživatel pak musí sám ošetřit hodnoty, které do registrů bude zapisovat. Např. při přímém zápisu proměnné typu *uint\_32t* na adresu se zapíše celkem 4 byty, což znamená, že na uživatelem definovanou adresu se zapíše spodní dva byty a na adresu o jedna vyšší se uloží zbylé dva byty. Toto chování je zcela automatické a je potřeba s ním počítat.

Struktura *mlc\_info\_t* v sobě uchovává verze firmware CPLD, FPGA a verzi hardwaru (revizi desky plošného spoje). Pokud je k MLC interface připojen externí modul, tak je vyplněna i verze jeho firmware. Tyto hodnoty jsou vyplněny při vykonání funkce *MLC\_init()*. Proměnná *adc\_conf* má v sobě uloženo konfigurační slovo pro ADC, uživatel může zpětně zjistit, jak má nastaveny analogově-digitální převodníky.

Ovladač lze rozdělit na několik nezávislých sekcí, z nichž každá abstrahuje konkrétní část hardwaru MLC interface. Aby bylo pro uživatele zcela zřejmé použití jednotlivých funkcí, byl zvolen unifikovaný (v rámci *MLC\_driver\_lib*) systém pojmenování. Všechny funkce náležející ovladači začínají předponou *MLC\_*, následuje zkratka hardwarové periferie viz Tab. 3.9 a poté přibližná funkce daného volání. Není-li uvedena zkratka periferie, jedná se o obecnou funkci, která není vázána na konkrétní hardwarový blok.

Tab. 3.9.: Zkratky použité pro jednotlivé periferie dostupné na MLC interface

Periferie	Zkratka
Analogově digitální převodník	ADC[n]_
Sériová USB linka	SCI_
Digitálně analogový převodník	DAC_
Výkonové výstupy CPLD	PWR_
Univerzální vstupy/výstupy CPLD	UIO_
LCD displej	LCD_
Přímý přístup do paměti	DMA_
Ladicí výstupy CPLD	DBG_

```
void MLC_init();  
void MLC_global_enable();  
void MLC_global_disable();
```

Výp. 3.2: Obecné funkce MLC\_interface\_lib

```
void MLC_ADC_reset(void);  
void MLC_ADC_setup(uint16_t os_1, uint16_t os_2, uint16_t os_3, uint16_t rng_1,  
    uint16_t rng_2, uint16_t rng_3);  
void MLC_ADC_setup_SOC(uint16_t adsoc);
```

Výp. 3.3: Funkce pro nastavení ADC

### 3.5.5.2. Obecné funkce MLC API

Sada obecných funkcí (viz Výp. 3.2) slouží k ovládání MLC interface jako celku. *MLC\_init()* provádí základní inicializaci mikrokontroléru, jeho hodin, vstupně výstupních pinů a provede základní inicializaci všech základních periférií včetně nastavení časování externí datové/adresové sběrnice. Zároveň tato funkce načítá do vnitřní struktury informace o firmwaru v CPLD a FPGA a také provádí detekci, zda je připojen externí modul rozšiřující počet PWM výstupů (viz kapitola 3.6.2). Uživatel tak může detekovat, jestli je jeho kód pro mikrokontrolér kompatibilní s nahráním firmwaru a hardwarovou verzí MLC interface. Tuto funkci uživatel musí zavolat jako první před použitím jakékoli další funkce z knihovny *MLC\_interface\_lib*.

Funkce *MLC\_global\_enable()* nastavuje globální signál *GLOBAL\_ENABLE* do log. 0, ten je primárně použit pro detekci resetu mikrokontroléru. Je zde využito faktu, že po resetu má mikrokontrolér piny nastaveny jako vstup s pull-up odpory, proto je po resetu, a to i softwarově (např. z Code Composer Studia přes JTAG rozhraní) signál *GLOBAL\_ENABLE* automaticky v log. 1. Základní firmware v CPLD i FPGA tento signál používá k blokování svých výstupů, CPLD vypne své PWR výstupy pro ovládání relé a FPGA nastaví své PWM výstupy do log. 0, pokud je tento signál v log. 1.

Zablokovat PWM výstupy z FPGA a PWR výstupy CPLD je možné pomocí funkce *MLC\_global\_disable()* programově, např. při vybavení ochran.

### 3.5.5.3. Funkce pro ovládání analogově-digitálních převodníků

Funkce pro ovládání ADC začínají (podle výše uvedeného) prefixem *MLC\_ADC\_*, případně *MLC\_ADCx\_*, kde *x* označuje konkrétní analogově digitální převodník (1 až 3).

Základní nastavení parametrů ADC se provádí po jejich resetu pomocí funkce *MLC\_ADC\_reset()* funkcí *MLC\_ADC\_setup(...)*, která nastavuje převzorkování a vstupní rozsah každého ze tří převodníků. Analogově-digitální převodníky mají každý svůj signál pro spuštění převodu, ale pomocí funkce *MLC\_ADC\_setup\_SOC(...)* lze nastavit i společné spuštění převodu. Plná definice funkcí viz (Výp. 3.3)

```
void MLC_ADC_start_conv(void);  
void MLC_ADC_start_one_conv(uint16_t channel);  
uint16_t MLC_ADC_wait(void);
```

Výp. 3.4: Funkce pro start převodu ADC

```
volatile int16_t* MLC_ADCx_read(void);  
volatile int16_t* MLC_ADCx_get_res_ptr(void);  
volatile mlc_adc_result* MLC_ADCx_get_res_strptr(void);
```

Výp. 3.5: Funkce pro čtení hodnot z ADC

Spustit převod je možné nejen hardwarově (např. PWM modulem), ale i softwarově. K tomu slouží dvě funkce (Výp. 3.4). *MLC\_ADC\_start\_conv()* spustí převod všech tří převodníků, zatímco u *MLC\_ADC\_start\_one\_conv(...)* uživatel volí konkrétní převodník, který má začít měřit. Pomocí volání *MLC\_ADC\_wait()* je možné programově počkat na dokončení převodu převodníků.

Následují funkce pro čtení dat z převodníků (Výp. 3.5). *MLC\_ADCx\_read()* programově čte data z analogově-digitálních převodníků a vrací ukazatel na pole načtených hodnot. Je možné načítat data z ADC pomocí DMA, pak jsou použity funkce *MLC\_ADCx\_get\_res\_ptr()* nebo *MLC\_ADCx\_get\_res\_strptr()*. Ty nevyčítají žádná data z ADC, počítají s tím, že ta již byla přenesena DMA přenosem, a vracejí pouze ukazatele na pole s hodnotami nebo ukazatel na strukturu obsahující hodnoty. Pole hodnot má velikost 16 slov, dvě slova vždy patří jednomu vstupu převodníku. Sudý index je horní část 18-ti bitového čísla a na lichých indexech jsou zbylé dva bity posunuté o 14 bitů doleva. Struktura s hodnotami převodu (Výp. 3.6) poskytuje uživateli přehlednější přístup k naměřeným hodnotám.

Funkce pro analogově digitální převodníky také podporují jednoduché nastavení přerušovací rutiny po dokončení převodu. Uživatel ke konfiguraci obsluhy přerušení použije funkci *MLC\_ADC\_enable\_isr(...)*, která zajišťuje kompletní nastavení mikrokontroléru. Voláním *MLC\_ADC\_disable\_isr()* lze volání přerušovací rutiny případně zakázat (viz Výp. 3.7).

#### 3.5.5.4. Funkce pro obsluhu DAC

MLC interface obsahuje osmikanálový digitálně-analogový převodník sloužící např. pro zobrazení vnitřních proměnných programu na osciloskopu. Převodník je připojen přes SPI sběr-

```
typedef struct{  
    int16_t in0;           /*!< ADC input 0 high word of result.*/  
    int16_t in0_low;      /*!< ADC input 0 low word of result.*/  
    ...  
    int16_t in8;           /*!< ADC input 7 high word of result.*/  
    int16_t in8_low;      /*!< ADC input 7 low word of result.*/  
} mlc_adc_result; /*!< Type definition of ADC result structure.*/
```

Výp. 3.6: Struktura pro ukládání dat z ADC



```
void MLC_ADC_enable_isr(interrupt void* handler);  
void MLC_ADC_disable_isr(void);
```

Výp. 3.7: Funkce pro nastavení obsluhy přerušení od ADC

```
uint16_t dac_values[8] = {0};  
uint16_t dac_sel_ch[8];  
void MLC_DAC_write(uint16_t count);  
void MLC_DAC_init(void);
```

Výp. 3.8: Funkce pro ovládání DAC

nici s taktem 20 MHz, funkce pro jeho obsluhu viz Výp. 3.8. Uživatel má k dispozici dvojici polí po osmi znacích. Hodnoty v poli *dac\_sel\_ch* určují mapování hodnot z pole *dac\_values* na kanály DAC. Např. je-li v *dac\_sel\_ch[0]* zapsána hodnota 2, pak se hodnota uložená v *dac\_values[0]* zapíše na třetí výstupní kanál DAC označený na základní desce písmenem C.

Samotný výpis na DAC provádí funkce *MLC\_DAC\_write(...)*, kdy její parametr určuje kolik hodnot z pole *dac\_values* počínaje indexem 0 se do DAC zapíše. SPI periferie používá svou FIFO o velikosti 16 bytů, výpis na DAC neblokuje mikrokontrolér během odesílání dat. Hodinová frekvence SPI periferie je nastavena na 20 MHz, jednotlivé hodnoty se na výstupu DAC nastavují s minimálním zpožděním i vzájemným časovým posuvem.

### 3.5.5.5. Funkce pro obsluhu SCI

Pro zjednodušení přístupu k sériové lince poskytované MLC interface byly implementovány podpůrné funkce pro tuto periferii procesoru (Výp.3.9). Funkcí *MLC\_SCI\_init(...)* uživatel nastaví požadovanou rychlost komunikace, funkce vypočte potřebné dělicí poměry pro SCI periferii, nastaví GPIO a FIFO. Pak lze voláním funkcí *MLC\_SCI\_send\_char(...)* a *MLC\_SCI\_send\_str(...)* odesílat znaky ale i znakové řetězce. Je-li z nějakého důvodu požadováno, aby funkce odesílání počkala na odeslání znaku, použije se varianta volání *void MLC\_SCI\_send\_char\_wait(...)*, která počká na vyprázdnění odesílací FIFO.

Obdobnou funkci má *MLC\_SCI\_wait\_send()*, která vrátí 0, pokud jsou ve FIFO ještě nějaké znaky k odeslání. Znak lze přijímat voláním *MLC\_SCI\_recv\_char()*. Funkce *MLC\_SCI\_char\_avail()* vrací uživateli počet přijatých znaků v přijímací FIFO SCI periferie.

```
void MLC_SCI_init(uint32_t baudrate);  
void MLC_SCI_send_char(char c);  
void MLC_SCI_send_str(char* str);  
void MLC_SCI_send_char_wait(char c);  
char MLC_SCI_recv_char(void);  
char MLC_SCI_char_avail(void);  
char MLC_SCI_wait_send(void);
```

Výp. 3.9: Funkce pro ovládání SCI



```
void MLC_DMA_enable_isr(uint16_t channel, interrupt void* handler);
void MLC_DMA_disable_isr(uint16_t channel);
void MLC_DMA_activate(void);
void MLC_DMA_deactivate(void);
uint16_t MLC_DMA_active(void);
```

Výp. 3.10: Funkce pro ovládání DMA

```
void MLC_LCD_clrscr(void);
void MLC_LCD_write_chr(char chr, uint16_t pos);
void MLC_LCD_write_str(char* pStr, uint16_t pos);
```

Výp. 3.11: Funkce pro ovládání LCD

### 3.5.5.6. Funkce pro DMA

Vzhledem k počtu přenášených dat z AD převodníků (48 bytů), je vhodné přenášet je pomocí přímého přístupu do paměti (DMA). V porovnání s programovým přenosem dojde ke zkrácení doby nutné na přenos naměřených dat. DMA také nezatěžuje jádro mikrokontroléru. Funkce pro práci s DMA jsou podporovány pouze u mikrokontrolérů TMS320F28335 a TMS570LS3137, mikrokontrolér TMS320F2812 nedisponuje DMA řadičem. Přenos dat do paměti mikrokontroléru vždy probíhá v sekvenci: data prvního (na hlavní desce označen AD CH1), potom ze druhého a nakonec z třetího ADC. Seznam dostupných volání ukazuje Výp. 3.10. Funkce *MLC\_DMA\_enable\_isr()* povoluje obsluhu přerušení od daného DMA kanálu (1 až 3), je tedy možné data zpracovávat ve třech dávkách, případně, pokud nejsou použity všechny tři AD převodníky, je možné přenášet pomocí DMA data jen z těch použitých. Funkcí *MLC\_DMA\_disable\_isr()* lze obsluhu přerušení zakázat. DMA přenosy je nutné pro použití ještě povolit i globálně voláním *MLC\_DMA\_activate()*, nebo je lze zakázat voláním funkce *MLC\_DMA\_deactivate()*. Poslední z funkcí DMA řadiče je *MLC\_DMA\_active()*, která vrací kladné číslo reprezentující momentálně aktivní (ve smyslu přenášení dat) DMA kanál. Návrátová hodnota 0 pak říká, že žádný DMA přenos aktuálně neprobíhá.

### 3.5.5.7. Funkce pro LCD

Stejně jako periferie SCI i LCD displej má implementovány podpůrné funkce v MLC API (Výp. 3.11). Volání *MLC\_LCD\_clrscr()* vypíše na displej nebo spíše do jeho vyrovnávací paměti samé mezery. *MLC\_LCD\_write\_chr()* umožňuje uživateli zapsat na přesnou pozici na displeji požadovaný znak. Obdobně *MLC\_LCD\_write\_str()* zapisuje řetězec znaků od uživatelem zadané pozice, pokud je vypisovaný řetězec delší než je počet znaků displeje, tak se řetězec na displeji začne přepisovat sám sebou.

## 3.6. Moduly pro MLC interface

### 3.6.1. Mikrokontrolérové moduly pro MLC interface

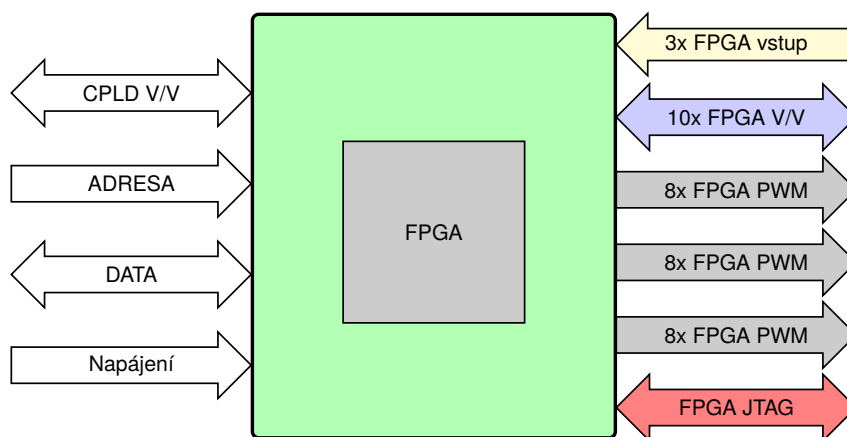
Obecně lze s MLC interface používat jakýkoli mikrokontrolér nebo procesor, který má vyvedenu 16-ti bitovou datovou a alespoň 9 bitů širokou adresovou sběrnici. Modul musí použitému mikrokontroléru/procesoru zajistit požadované napájení, včetně jeho korektního náběhu a musí provádět jeho korektní reset po zapnutí napájení. Reset signál pro procesor je dále vyveden na MLC interface jako systémový reset a využívají jej systémové CPLD i uživatelské FPGA. Byly vyvinuty celkem tři mikrokontrolérové moduly s obvody TMS320F2812 [61], TMS320F28335 [62] a TMS570LS3137 [63].

Konektory modulu jsou zapojeny tak, aby byly vývodově kompatibilní s kitem *eZdsp 28335* firmy Spectrum Digital [49], je tak možné použít i tyto komerčně dostupné kity spolu s MLC interface. Vyvinuté moduly nevyužívají patiči pro procesor a výrazně se tak zvyšuje jejich spolehlivost na úkor jednoduché vyměnitelnosti mikrokontroléru na modulu. Vyvinuté moduly neintegrují JTAG emulátor (na rozdíl od výše zmíněného *eZdsp* kitu), tím se podstatně zkrátí délka JTAG sběrnice a počet obvodů na ní napojených, což se příznivě projevilo na její odolnosti vůči EMI. Všechny moduly poskytují možnost osadit galvanicky izolovaný budič CAN sběrnice s integrovaným DC-DC měničem, založený na obvodu ADM3053 [64] firmy Analog Devices. Fotografie jednotlivých modulů jsou v Příloze A.

### 3.6.2. PWM modul pro MLC interface

Při analýze nároků na řídicí platformu pro řízení 7L-CHB měniče bylo zjištěno, že MLC interface není schopen poskytnout potřebných 72 PWM signálů pro řízení třífázové varianty této topologie. Jedním z řešení použitelných pro tuto situaci je rozšíření počtu PWM signálů jejich rozdělením, jak je popsáno v kapitole 4.4 této práce. Nabízí se i použití dvou regulátorů MLC interface pro řízení takto komplexního měniče. Alternativní cestou je využít přídatný modul (na jehož použití je hlavní deska MLC interface připravena), který požadovanou funkcionalitu doplní.

Nejjednodušeji, obvodově i z hlediska programové obsluhy, a také cenově nejvýhodněji vychází realizace doplňkového modulu pro MLC interface s FPGA obvodem. V porovnání s řešením založeném na dvou MLC interface je doplňkový modul řádově levnější a není nutno řešit synchronizaci běhu mezi dvěma číslicovými regulátory. Na rozdíl od řešení s rozdělením signálů je zde hlavní výhodou ovládání každého signálu z jednoho pinu FPGA, měnič, resp. jeho výkonové prvky jsou tak více pod kontrolou uživatele. Modul byl vyvinut jako přídatná deska plošných spojů, která se připojí přímo na konektory pro přídatné moduly MLC interface, které zajistí její napájení a napojení na systémovou paralelní datovou a adresovou sběrnici pomocí níž je s modulem komunikováno. Vyvedené sběrnice naznačuje blokové



Obr. 3.22.: Blokové schéma přídatného PWM modulu

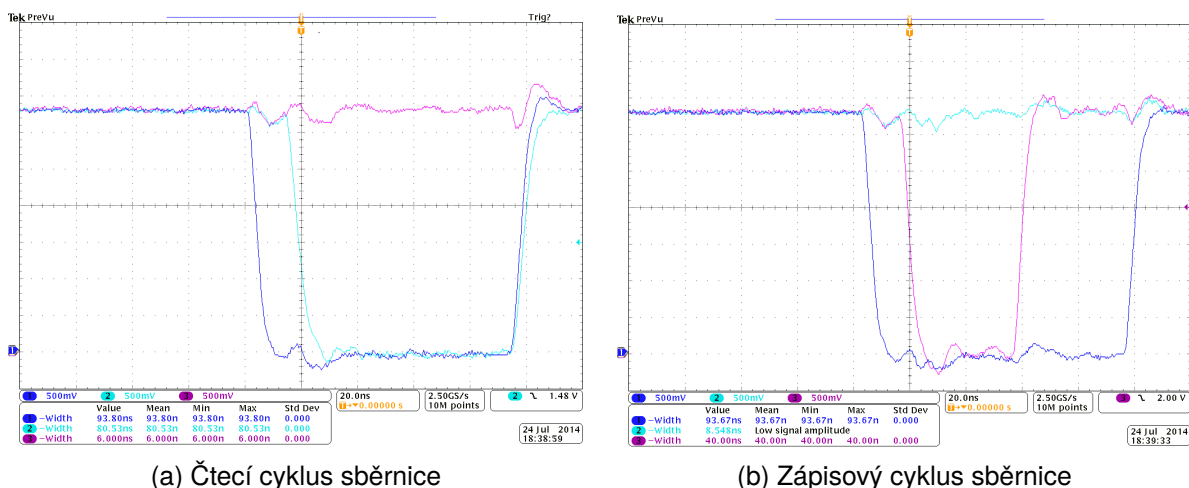
schéma na Obr. 3.22. Systémová propojení mají bílá pozadí, barevně jsou naznačeny uživatelské vstupy a výstupy.

Způsob propojení FPGA se zbytkem MLC interface je tedy stejný jako u FPGA osazeného přímo na MLC interface. Modul má osazeno FPGA výrobce Altera, typ EP3C10K, ten poskytuje dostatek prostředků i vývodů pro doplnění chybějících 24 PWM výstupů s přizpůsobovací úrovní (šedé pozadí na Obr. 3.22). Zbylé vývody jsou vyvedeny na konektor vývodově kompatibilní s PWM konektory, avšak je zde vynechán obvod pro přizpůsobení úrovní, tím lze tyto vývody provozovat na vyšších frekvencích a budou využity pro realizaci PWM modulátorů s vysokou spínací frekvencí v řádu stovek kHz pro měniče s měkkou komutací nebo pro vysokorychlostní komunikaci v reálném čase s nadřazenou řídicí jednotkou. Fotografie modulu je uvedena v příloze A, Obr. A.5 a v sestavě spolu s MLC interface na Obr. A.6.

### 3.7. Výsledky měření kritických parametrů MLC interface

Mezi kritické parametry MLC interface patří rychlost přenosové sběrnice, ta ovlivňuje veškerou činnost algoritmů implementovaných do mikrokontroléru i FPGA. Sběrnice vkládá dopravní zpoždění, se kterým je potřeba počítat i v regulačních algoritmech.

Bylo provedeno měření latencí a dosažitelné přenosové rychlosti mezi programovatelnými obvody a periferiemi. K tomuto testování byl použit modul s mikrokontrolérem TMS320F28335, protože je v současnosti nejpoužívanější pro implementaci regulačních algoritmů. Oscilogramy na Obr. 3.23a a 3.23b ukazují výsledné průběhy. Z nich lze vyčíst latenci externí sběrnice  $93\text{ ns}$  (přechod  $\overline{\text{CS}}$  signálu  $H \rightarrow L \rightarrow H$ ) pro zápis i čtení. Mikrokontrolér TMS320F28335 vkládá mezi každý přenos dalších šest svých hodinových cyklů, což prodlužuje cyklus sběrnice o dalších  $40\text{ ns}$ . Přenáší se 16 bitů a tak přenosová rychlost je  $15\text{ MB/s}$ .



Obr. 3.23.: Cyklus externí sběrnice TMS320F28335 a MLC interface, CH1 (modrá):  $\overline{CS}$  signál externích periferií, CH2 (světle modrá): signál RD, CH3 (fialová): signál WR

### 3.7.1. Měřící cyklus analogově-digitálních převodníků

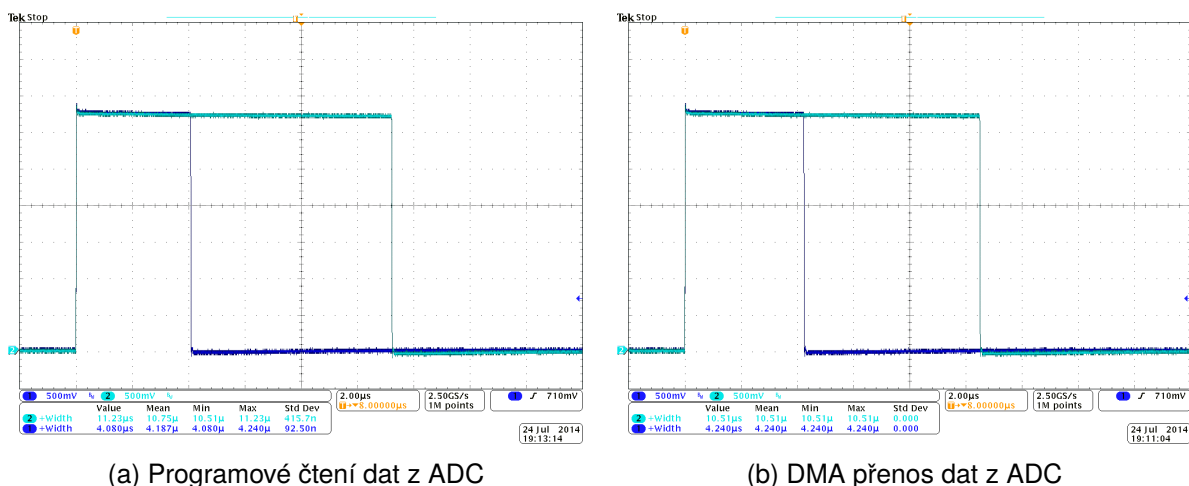
Klíčovou komponentou pro implementaci jakékoli regulace je vstup měřených analogových hodnot, v případě řízení výkonové elektroniky je důležitá latence analogově-digitálních převodníků, tedy doba mezi změněním hodnoty a její dostupnosti pro řídicí algoritmus. Knihovna MLC\_interface\_lib podporuje dva přístupy k vyčítání hodnot z AD převodníků, programové a pomocí přímého přístupu do paměti. Čtení hodnot pomocí programové smyčky vykonávané mikrokontrolérem bylo implementováno kvůli mikrokontroléru TMS320F2812, který nedisponuje řadičem DMA. Celkovou dobu převodu a přenosu dat lze vypočítat ze vzorce 3.1.

$$T_{ADC} = T_{CONV} + N_{ADC} \cdot N_{samples} \cdot T_{BUS} \quad (3.1)$$

kde

- $T_{ADC}$  ... celková doba převodu a přenosu dat
- $T_{CONV}$  ... doba převodu ADC
- $N_{ADC}$  ... počet převodníků
- $N_{samples}$  ... počet vzorků naměřených jedním převodníkem
- $T_{BUS}$  ... délka cyklu sběrnice

Oscilogramy na Obr. 3.24a a 3.24b porovnávají obě implementace. Z oscilogramů je patrné zkrácení měřícího cyklu ADC při přenosu dat pomocí DMA z  $11,23 \mu s$  na  $10,51 \mu s$ , respektive zkrácení samotného přenosu dat z  $7,15 \mu s$  na  $6,27 \mu s$ . DMA přenos navíc nespótrebovává procesorový čas, zbývá tak více času na samotné výpočty. Přiřazením unikátní adresy každému z převodníků lze rozdělit DMA přenos na tři navazující přenosy, tím je pak možné analogové hodnoty citlivé na dopravní zpoždění zpracovat s jeho menší hodnotou.



Obr. 3.24.: Čtecí cyklus externích A/D převodníků, CH1 (modrá): doba převodu ( $T_{CONV}$ ), CH2 (světle modrá): doba převodu a přenosu dat ( $T_{ADC}$ )

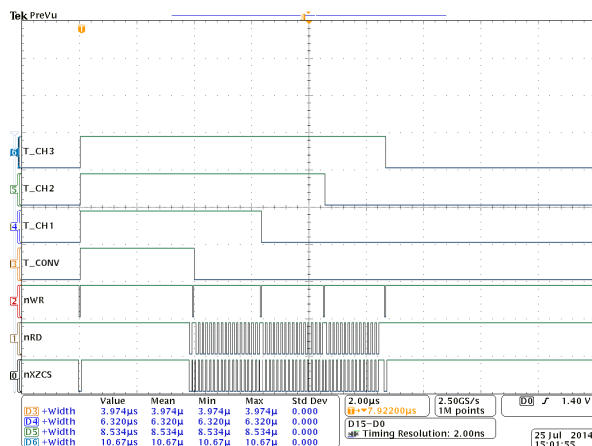
Průběhy osvětlující tento princip viz Obr. 3.25. Signál označený  $T_{CONV}$  indikuje aktivní převod analogově digitálních převodníků,  $T_{CH1}$  zobrazuje zpoždění obsluhy přerušovací rutiny pro převodník č. 1 od spuštění převodu (tj. včetně DMA přenosu dat do paměti mikrokontroléru), obdobně  $T_{CH2}$  a  $T_{CH3}$  zobrazují zpoždění vyvolání přerušovacích rutin pro převodník č. 2 resp. č. 3. Každý ze signálů  $T_{CHx}$  má možnost vyvolat přerušení, tzn. spustit rutinu pro zpracování naměřených dat. Signály označené  $nWR$ ,  $nRD$  a  $nXZCS0$  jsou řídicí signály paralelní sběrnice na MLC interface.

### 3.8. Porovnání implementací řízení měniče 4L-FLC z pohledu využitých prostředků

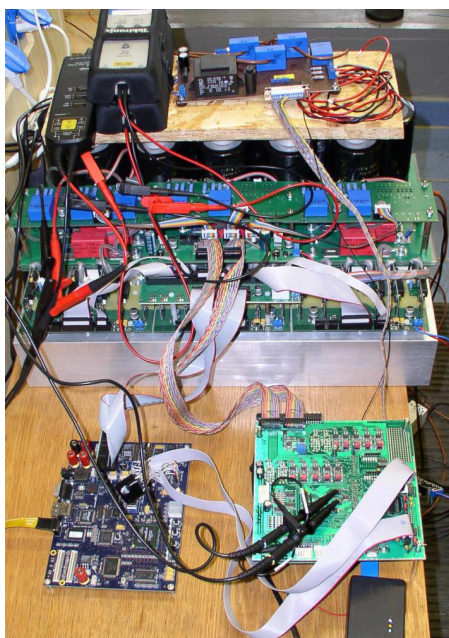
V tomto odstavci je uvedeno základní porovnání výkonu a vlastností MLC interface s původním řídicím systémem, který sestával z původního mikroprocesorového regulátoru a vývojového kitu s FPGA Cyclone II.

Pro srovnání byla vybrána aplikace čtyřúrovňového měniče s plovoucími kondenzátory (4L-FLC), kde je FPGA použito pro implementaci balancování napětí na plovoucích kondenzátorech

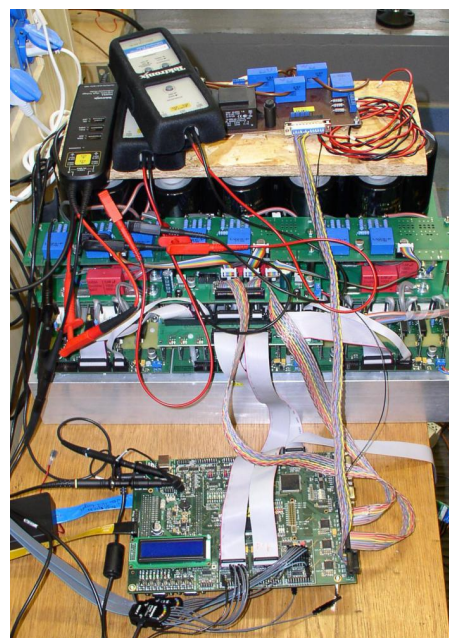
Pro první experimenty s měničem 4L-FLC, kdy byla použita balancovací tabulka, bylo nutné použít hradlové pole pro její implementaci. Původní číslicový regulátor nevyužívá adresovou ani datovou sběrnici mikrokontroléru, propojení s FPGA kitem tedy bylo realizováno přes PWM brány a byl navržen komunikační protokol, který umožňuje do entit v FPGA přenést data nutná pro řízení měniče. Výsledné zapojení je na Obr. 3.26a. Na Obr. 3.26b je vidět náhrada původního řešení pomocí vyvinutého mikroprocesorového regulátoru MLC interface.



Obr. 3.25.: DMA datový přenos - možnost postupného zpracování naměřených hodnot, CH0 (černá): signál  $\overline{CS}$  externích periférií, CH1 (hnědá): signál  $\overline{RD}$ , CH2 (červená): signál  $\overline{WR}$ , CH3 (oranžová): doba převodu převodníků  $T_{CONV}$ , CH4 (modrá): doba převodu a přenosu dat převodníku č.1, CH5 (zelená): doba převodu a přenosu dat převodníku č.2, CH6 (světle modrá): doba převodu a přenosu dat převodníku č.3



(a) Střídač s 4L-FLC řízený pomocí kombinace vývojového kitu s TMS32F28335 a Cyclone II



(b) Střídač s 4L-FLC řízený pomocí MLC interfa-  
ce

Obr. 3.26.: Porovnání realizací řídicích platform pro 4L-FLC

Tab. 3.10.: Srovnání vlastností analogových vstupů dvou generací číslicového regulátoru

Parametr	Interface + FPGA kit	MLC interface
rozlišení ADC	12 bitů	18 bitů
Doba konverze + přenosu dat	80 ns + 26 ns	4 + 2,4 μs
ADC vstupů	14	24
Počet současně samplovaných vstupů	2	24

Tab. 3.11.: Srovnání využití zdrojů dvou generací vývojových platform

Parametr	Interface + FPGA kit	MLC interface
Počet použitých logických elementů	1763	1403
Úspora velikosti kódu v mikrokontroléru	0	120 bytů
Doba přenosu dat z mikrokontroléru do FPGA	10 μs	330 ns

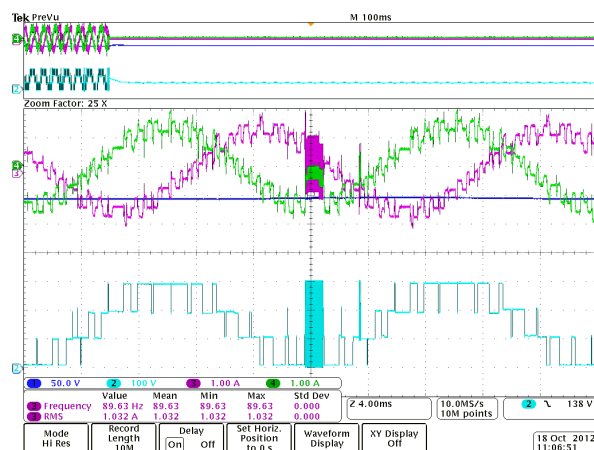
Obě dvě zapojení jsou funkčně ekvivalentní a lze je tedy porovnat v klíčových parametrech, které jsou shrnuty v tabulkách Tab 3.10 a 3.11. Z tabulky Tab. 3.10 plyne, že hlavní devizou MLC interface je rychlá komunikace s FPGA a počet v jeden okamžik samplovaných analogových vstupů, je však pomalejší z pohledu rychlosti převodu analogově-digitálních převodníků a následném přenosu naměřených dat do paměti mikrokontroléru. Nicméně použité externí AD převodníky jsou ve srovnání s těmi na čipu mikrokontroléru mnohem přesnější a odolnější proti rušení vznikajícího činností digitálních obvodů. Z pohledu využití systémových prostředků je lepší MLC interface, kdy odpadá implementace komunikačního protokolu, což se projeví úsporou nejen procesorového času, ale i logických elementů v hradlovém poli (viz Tab 3.11).

Řešení s dvěma vývojovými kity se navíc ukázalo obtížně použitelné v okamžiku, kdy napětí meziobvodu  $U_{dc}$  přesáhlo 200 V stejnosměrných. Po překročení tohoto napětí se začaly objevovat oscilace výstupního napětí měniče (Obr. 3.27). Ty způsobovala kombinace nízké vstupní impedance řídicích obvodů měniče spolu s dosažitelným malým výstupním proudem a napětím z hradlového pole, které nedisponovalo žádnými posilovacími budiči. Řešení problému s rušením nízkonapěťových řídicích signálů je podrobněji rozvedeno v sekci 4.4.1.

### 3.9. Zhodnocení základní vývojové platformy MLC interface a přidružených nástrojů

MLC interface je založen na stávajících zkušenostech s provozem číslicových regulátorů a vývojových kitů naší i komerční výroby. Kombinace mikrokontroléru a FPGA, na které je celá topologie založena umožňují vývoj a realizaci řízení složitých topologií výkonových polo-





Obr. 3.27.: Oscilace výstupního napětí měniče 4L-FLC způsobené rušením řídicích signálů, RL zátěž,  $U_{dc}=300$  V, CH2 (světle modrá): fázové napětí střídače  $u_{iu}$  [100 V/d], CH3 (fialová): fázový proud střídače  $i_{iu}$  [1 A/d], CH4 (zelená): fázový proud střídače  $i_{iv}$  [1 A/d]

vodičových měničů a přenesení částí algoritmů z mikrokontroléru do hradlového pole (např. balancování napětí na kondenzátorech) ušetří výpočetní čas mikrokontroléru pro implementaci složitějších algoritmů.

Modularita mikroprocesorového regulátoru MLC interface umožňuje jednoduché využití libovolných mikrokontrolérů s externí adresovou a datovou sběrnici. Možnost připojení dalších modulů přímo na sběrnici MLC interface a mikrokontroléru zlepšuje možnost adaptace na úkoly s nimiž při prvotním návrhu nebylo počítáno.

Vstupy a výstupy dostupné na MLC interface dostačují pro řízení až devítiúrovňových měničů v třífázové variantě napěťového střídače nebo pulzního usměrňovače. Řízení kompletního (tzn. pulzní usměrňovač a střídač) nepřímého měniče frekvence je možné až do pětiúrovňové varianty bez dalších přídavných zařízení. Pokud by bylo využito technologie rozbočení PWM signálu pro komplementární prvky, bylo by možné řídit až devítiúrovňový kompletní nepřímý měnič kmitočtu. Přidáním relativně jednoduchého modulu s dalším FPGA obvodem, je možné přímo (bez rozbočování PWM signálů) řídit kompletní měnič kmitočtu o sedmi úrovních.

Ladicí nástroje jsou zastoupeny galvanicky oddělenými JTAG emulátory připojovanými přes USB sběrnici. Ty jsou navrženy tak, aby odolávaly elektromagnetickému rušení vznikajícím při provozu měničů, a bylo tak zajištěno spolehlivé ladění algoritmů řízení při plném zatížení výkonové části. Jejich univerzální návrh a flexibilita v kombinaci s externím programem umožňuje programovat (bez možnosti ladění) hradlová pole, která jsou součástí vývojového kitu.



## 4. Vývoj PWM modulátorů pro vícehladinové měniče středního a velkého výkonu

Víceúrovňové měniče jsou perspektivní cestou jak obejít napěťová omezení současně dostupných výkonových polovodičových součástek, zároveň mají větší možnosti v oblasti modulace výstupního napětí a tím snižují vyzařované elektromagnetické rušení i zastoupení harmonických ve výstupním signálu. Spolu s nižším  $\frac{dU}{dt}$  se také snižuje namáhání izolace kabelů i motorů.

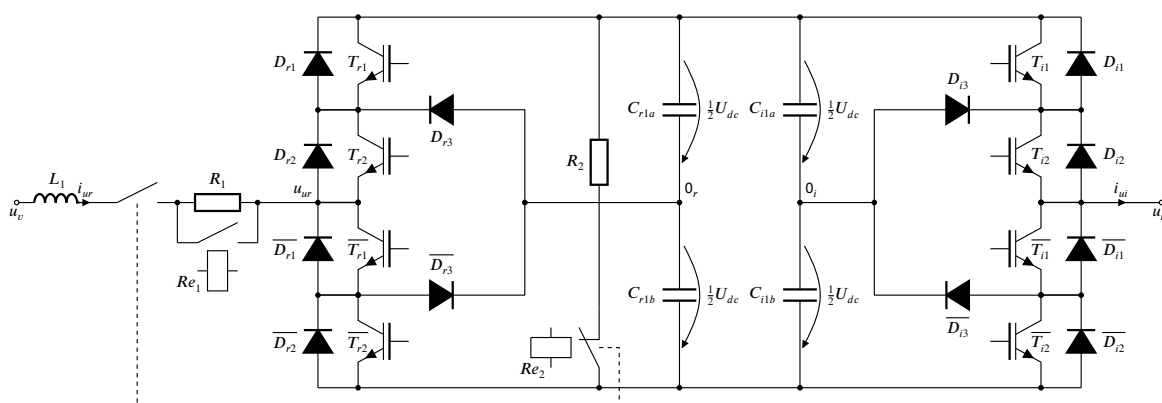
Zásadní nevýhodou jsou nároky na počet řídicích signálů pro výkonové prvky spolu s počtem analogových měření (to se týká především FLC měničů) a větší nároky na řízení těchto komplexních topologií. V této kapitole budou rozebrány základní vlastnosti víceúrovňových měničů, které autor zkoumal a vyvíjel v rámci této disertační práce.

### 4.1. Tříúrovňový měnič s upínacími diodami: 3L-NPC

Obecně jsou měniče s topologií NPC složeny z klasických dvojúrovňových měničů spojených do série, v našem případě jsou tyto měniče použity dva. Původní výstupy dvojúrovňových měničů (spojnice  $T_{x1} \leftrightarrow T_{x2}$  a  $\overline{T_{x1}} \leftrightarrow \overline{T_{x2}}$ ) jsou přes diody  $D_{x3}$  a  $\overline{D_{x3}}$  (viz Obr. 2.1a) propojeny s nulovým bodem, který vznikne sériovým spojením dvou kondenzátorů v meziobvodu a zároveň na spojnici  $T_{x2} \leftrightarrow \overline{T_{x1}}$  je dostupný "nový" fázový výstup měniče. Ze zapojení plyne, že kondenzátory dělí napětí meziobvodu na polovinu a tak každý z aktivních prvků blokuje pouze poloviční napětí. Nyní je také možné na výstupní svorku měniče připojit nejen  $\pm \frac{U_{dc}}{2}$ , ale i nulové napětí (viz Tab. 4.1).

Výkonové prvky jsou u tohoto měniče rozděleny do komplementárních dvojic, což je naznačeno ve schématu stejným indexem, ale nadtržením u prvku ovládaného invertovaným řídicím signálem.

Zapojení měniče 3L-NPC v laboratoři včetně podpůrných součástek viz Obr. 4.1. Pro přehlednost je na obrázku pouze jedna fáze střídače a usměrňovače, ty jsou se zbylými fázemi propojeny pomocí stejnosměrného meziobvodu. Start měniče je plně v režii mikroprocesorového regulátoru MLC interface, který pomocí svých PWR výstupů ovládá dvojici relé  $Re_1$  a  $Re_2$ , kdy je pomocí  $Re_2$  a  $R_1$  nabit stejnosměrný meziobvod měniče a po dokončení nabíjení se měnič připojí přímo na síť sepnutím  $Re_1$ . Obě relé mají použity tři kontakty, pro každou fázi měniče jeden.



Obr. 4.1.: Schéma jedné fáze 3L-NPC nepřímého měniče kmitočtu

Tab. 4.1.: Výstupní fázové napětí a vliv na kondenzátory střídače 3L-NPC pro jednotlivé kombinace sepnutí výkonových prvků a směr proudu

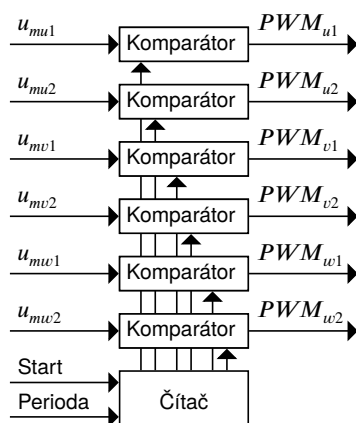
$T_{i1}$	$T_{i2}$	$u_{ui}$	$i_{ui}$	$C_{1a}$	$C_{1b}$	Hladina
0	0	$-\frac{U_{dc}}{2}$	-	-	-	1
0	1	0	$< 0$ $> 0$	vybíjí se nabíjí se	nabíjí se vybíjí se	2
1	1	$\frac{U_{dc}}{2}$	-	-	-	3

#### 4.1.1. PWM modulátor pro 3L-NPC

Pro správné odladění a ověření funkce vektorových modulátorů pro 3L-(A)NPC, byl implementován jednoduchý modulární generátor PWM signálů do FPGA. Hardware měniče využívá nízkourovňových kontrolérů s CPLD popsaných v sekci 4.4.1. PWM modulátor je tudíž jednoduchá sestava jednoho čítače s devíti komparačními registry, které generují jednobitový signál pro řízení jednotlivých komplementárních dvojic výkonových prvků měniče. Logika měniče zajišťuje rozdělení řídicích signálů na příslušné prvky a vložení mrtvých časů. Tyto modulátory jsou v hradlovém poli dva, pro pulzní usměřovač a střídač. Do stávajícího firmwaru FPGA je integrován podobným způsobem jako PD-PWM modulátor pro čtyřúrovňový měnič (viz Obr. 4.6) a jeho blokové schéma je zobrazeno na Obr. 4.2.

## 4.2. Čtyřúrovňový měnič s plovoucími kondenzátory: 4L-FLC

Topologie měničů s plovoucími kondenzátory je podobná NPC, avšak místo upínacích diod má struktura použity plovoucí kondenzátory (Obr. 4.4). Ty slouží stejně jako kondenzátory dělicí  $U_{dc}$  u NPC k rovnoměrnému rozložení tohoto napětí na výkonové součástky. Na rozdíl od NPC mají FLC měniče tzv. redundantní stavy, kdy lze na výstupu dosáhnout stejné hladiny napětí více kombinacemi sepnutých součástek, viz Tab. 4.2. Těto vlastnosti je využito pro udržení příslušných napětí na plovoucích kondenzátorech při provozu měniče.



Obr. 4.2.: Blokové schéma modulátoru pro 3L-NPC

Z pohledu fyzické realizace lze FLC měniče velmi dobře realizovat modulární formou a mít tak možnost počet úrovní přizpůsobit dané aplikaci. Situaci naznačuje Obr. 4.3. Jedna buňka je opět dvojúrovňový měnič (fialově označený), který má rozpojenu výstupní svorku a do ní je zapojena další buňka. V obrázku je zmíněna rovnice (4.1), s její pomocí je možné spočítat nominální napětí na jednotlivých buňkách, resp. jejich plovoucích kondenzátorech. Každý z kondenzátorů má tedy jiné pracovní napětí, avšak blokovací napětí polovodičových součástek všech buněk je stejné a rovné velikosti napětí na  $i$ -té buňce. To je rozdíl oproti NPC topologii, která má všechny kondenzátory na stejné napětí.

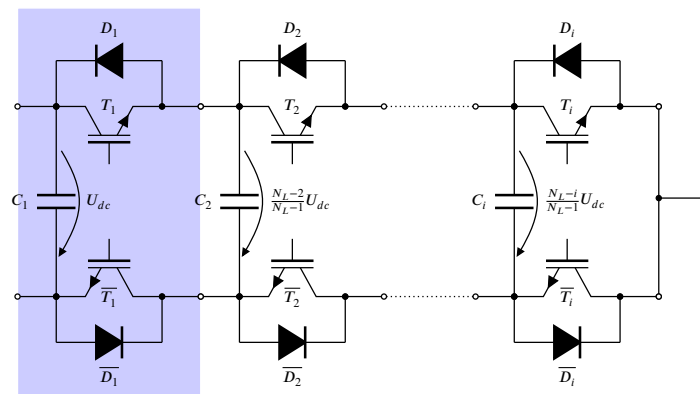
$$U_{C_i} = \frac{N_L - i}{N_L - 1} \cdot U_{dc} \quad (4.1)$$

kde

- $U_{C_i}$  ... napětí na kondenzátoru s indexem  $i$
- $N_L$  ... počet úrovní měniče
- $i$  ... index buňky měniče
- $U_{dc}$  ... napětí meziobvodu

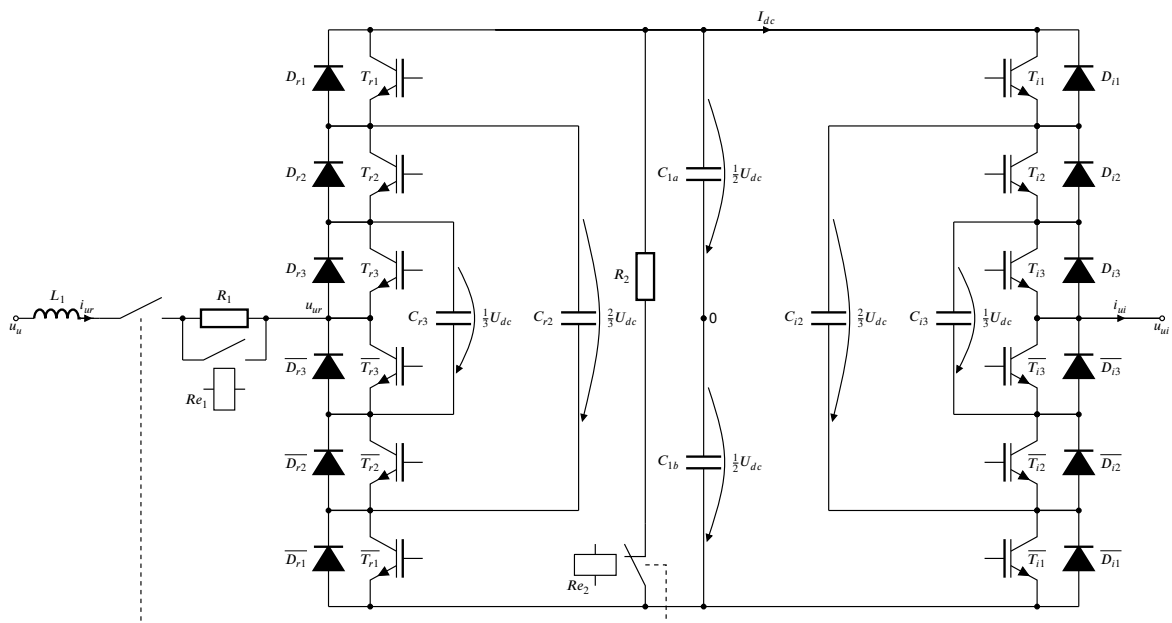
Výhodou FLC měničů je dosažitelný jakýkoli počet výstupních úrovní (liché i sudé). Nevýhodou je, že proud zátěží většinou teče přes kondenzátory, což na ně klade velké nároky z hlediska proudové zatížitelnosti, oteplení a spolehlivosti.

Topologie čtyřúrovňového měniče s plovoucími kondenzátory, tak jak byla postavena na prototypu v laboratoři je vykreslena na Obr. 4.4. Stejně jako u 3L-NPC, toto zapojení měniče umožňuje s pomocí MLC interface implementovat kompletní startovací proceduru. Pro přehlednost je vykreslena jen jedna fáze střídače a usměrňovače. S dalšími fázemi jsou propojeny přes stejnosměrný meziobvod.



Obr. 4.3.: Modularita FLC měničů

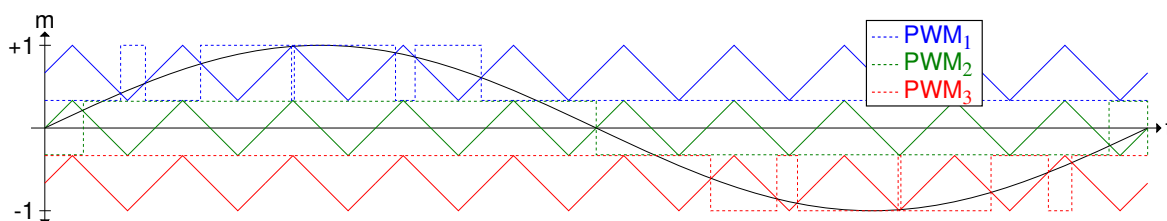
Jak plyne z názvu, tato topologie měniče umožňuje v rámci výstupního fázového napětí měniče pracovat se čtyřmi hladinami. Ze zapojení je zřejmé, že výkonové prvky pracují v komplementárním režimu, což je naznačeno v rámci označení IGBT, např.  $T_3$  je komplementární s  $\bar{T}_3$  atd. Až na výjimečné případy (jako je přednabíjení viz kapitola 5.2.1) nejsou tyto tranzistory sepnuty v jeden okamžik. Každá tato komplementární dvojice je přemostěna plovoucím kondenzátorem ( $C_2$  a  $C_3$ ). Tyto kondenzátory mají za úkol udržet na jednotlivých dvojicích prvků stejná napětí, v tomto konkrétním případě  $\frac{1}{3}U_{dc}$ . Podle sepnuté kombinace prvků se určuje výstupní napětí a v závislosti na směru proudu i nabíjení či vybíjení jednotlivých plovoucích kondenzátorů, toto pro střídač shrnuje Tab. 4.2. V případě usměrňovače by vyšla obdobná tabulka, avšak proud by se bral s opačným znaménkem. Všechna napětí jsou vztažena vůči virtuálnímu nulovému potenciálu na spoji kondenzátorů  $C_{1a}$  a  $C_{1b}$ .



Obr. 4.4.: Schéma jedné fáze 4L-FLC nepřímého měniče kmitočtu

Tab. 4.2.: Výstupní fázové napětí a vliv na plovoucí kondenzátory střídače 4L-FLC pro jednotlivé kombinace sepnutí výkonových prvků a směr proudu

$T_{i1}$	$T_{i2}$	$T_{i3}$	$u_{ui}$	$i_{ui}$	$C_{i2}$	$C_{i3}$	Hladina
0	0	0	$-\frac{U_{dc}}{2}$	-	-	-	1
0	0	1	$-\frac{U_{dc}}{6}$	< 0 > 0	- -	nabíjí se vybíjí se	2
0	1	0	$-\frac{U_{dc}}{6}$	< 0 > 0	nabíjí se vybíjí se	vybíjí se nabíjí se	2
0	1	1	$-\frac{U_{dc}}{6}$	< 0 > 0	nabíjí se vybíjí se	- -	2
1	0	0	$\frac{U_{dc}}{6}$	< 0 > 0	vybíjí se nabíjí se	- -	3
1	0	1	$\frac{U_{dc}}{6}$	< 0 > 0	vybíjí se nabíjí se	nabíjí se vybíjí se	3
1	1	0	$\frac{U_{dc}}{6}$	< 0 > 0	- -	vybíjí se nabíjí se	3
1	1	1	$\frac{U_{dc}}{2}$	-	-	-	4

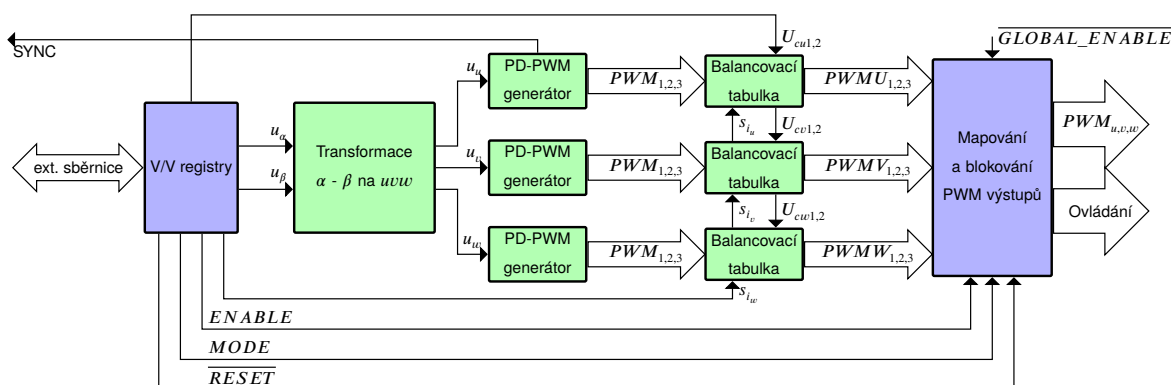


Obr. 4.5.: Princip PD-PWM modulace

Měnič má osazena čidla pro měření napětí na všech plovoucích kondenzátorech plus jedno v meziobvodu a dále každý vstupní i výstupní proud.

#### 4.2.1. PD-PWM modulátor pro 4L-FLC

Princip modulace PD-PWM spočívá v sestavení  $N - 1$  nosných pil PWM, které rozdělíme do  $N - 1$  sektorů umístěných nad sebou. Číslo  $N$  představuje počet hladin měniče, v našem případě je  $N = 4$ . Využívají se tedy celkem tři modulační pily spolu s jedním modulačním signálem (Obr. 4.5), kde modrá pila řídí dvojici  $T_{x1} + \overline{T_{x1}}$ , zelená  $T_{x2} + \overline{T_{x2}}$  a červená  $T_{x3} + \overline{T_{x3}}$ . Výsledné řídicí signály pro výkonové prvky však nejsou vhodné pro přímé řízení aktivních prvků, neboť tento typ modulace sám o sobě nezajišťuje balancování napětí na plovoucích kondenzátorech. Pro PD-PWM je tedy potřeba implementovat dekodér PWM signálů, v pracích [65, 66] jsou popsány některé z možných přístupů k řešení tohoto problému. Použité aktivní balancování je založeno na zavedení zpětné vazby od napětí na jednotlivých plovoucích kondenzátorech a směru proudu do modulátoru. Ten pak využívá redundantní stavy ve



Obr. 4.6.: Bloková struktura modulátoru pro 4L-FLC v FPGA

druhé a třetí hladině (viz Tab. 4.2) k udržování správného pracovního napětí na kondenzátorech.

Modulátor (Obr. 4.6) je implementován v FPGA, byl využit základní firmware (modře obarvený, viz sekce 3.5.3) a do něj byly jako uživatelské entity (zelené) přidány navržené PD-PWM modulátory s aktivním balancováním. Vstupem do modulátoru je požadovaný vektor výstupního napětí v  $\alpha$ - $\beta$  souřadnicích. Nejdříve je tento vektor rozložen do třífázového souřadného systému (u-v-w), jednotlivé složky jsou poté použity jako modulační signál v PD-PWM generátoru pro vytvoření PWM signálů pro komplementární dvojice tranzistorů. Tyto signály jsou v balancovací tabulce zkombinovány s příznaky polaritý proudu (na Obr. 4.6 označeny jako  $s_{i_u}$ ,  $s_{i_v}$ ,  $s_{i_w}$ ) a stavu nabití plovoucích kondenzátorů pro danou fázi (označeny  $U_{cv1,2}$ ,  $U_{cv1,2}$  a  $U_{cw1,2}$ ). Výsledkem jsou jednobitové ovládací signály pro jednotlivé komplementární dvojice 4L-FLC měniče. Vygenerované PWM vstupují do bloku mapování na konkrétní piny FPGA, kdy tento blok zajišťuje i ovládání měniče (přeposílá mu signály  $ENABLE$ ,  $MODE$  a  $RESET$  řídicí nízkourovňový kontrolér popsany v sekci 4.4) a případné odpojení výstupů na základě signálu  $GLOBAL\_ENABLE$ . Běh regulačních algoritmů v mikrokontroléru je synchronizován s během PD-PWM modulátoru pomocí synchronizačního signálu  $SYNC$ . Tento signál je generován jedním z PD-PWM generátorů. Ty jsou vzájemně synchronní a proto stačí  $SYNC$  generovat jen z jednoho.

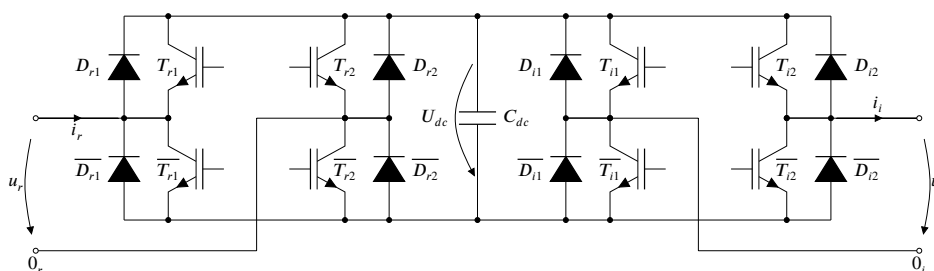
Algoritmus vyvažování je založen na principu dvouhodnotové regulace napětí na plovoucích kondenzátorech, neboť balancovací tabulka nepracuje s reálnými hodnotami napětí a proudů, ale mikrokontrolér do k tomu určenému registru zapisuje výše zmíněné příznaky reprezentované jako bitové proměnné. Ty jsou generovány mikrokontrolérem na základě jím naměřených hodnot proudů a napětí.

### 4.3. Sériově řazené H můstky: 7L-CHB

Další topologie měniče, kterou autor zkoumal a která je řízená pomocí MLC interface, je sedmiúrovňový měnič složený ze sériově řazených H můstků. Základní stavební buňka této

Tab. 4.3.: Výstupní fázové napětí a vliv na kondenzátor střídače s jedním H-můstkem pro jednotlivé kombinace sepnutí výkonových prvků a směr proudu

$T_{i1}$	$T_{i2}$	$u_i$	$i_i$	$C_{dc}$	Hladina
0	0	0	-	-	1
0	1	$U_{dc}$	$< 0$ $> 0$	nabíjí se vybíjí se	2
1	0	$-U_{dc}$	$< 0$ $> 0$	vybíjí se nabíjí se	3
1	1	0	-	-	4



Obr. 4.7.: Schéma jedné buňky nepřímého měniče kmitočtu s H můstkem

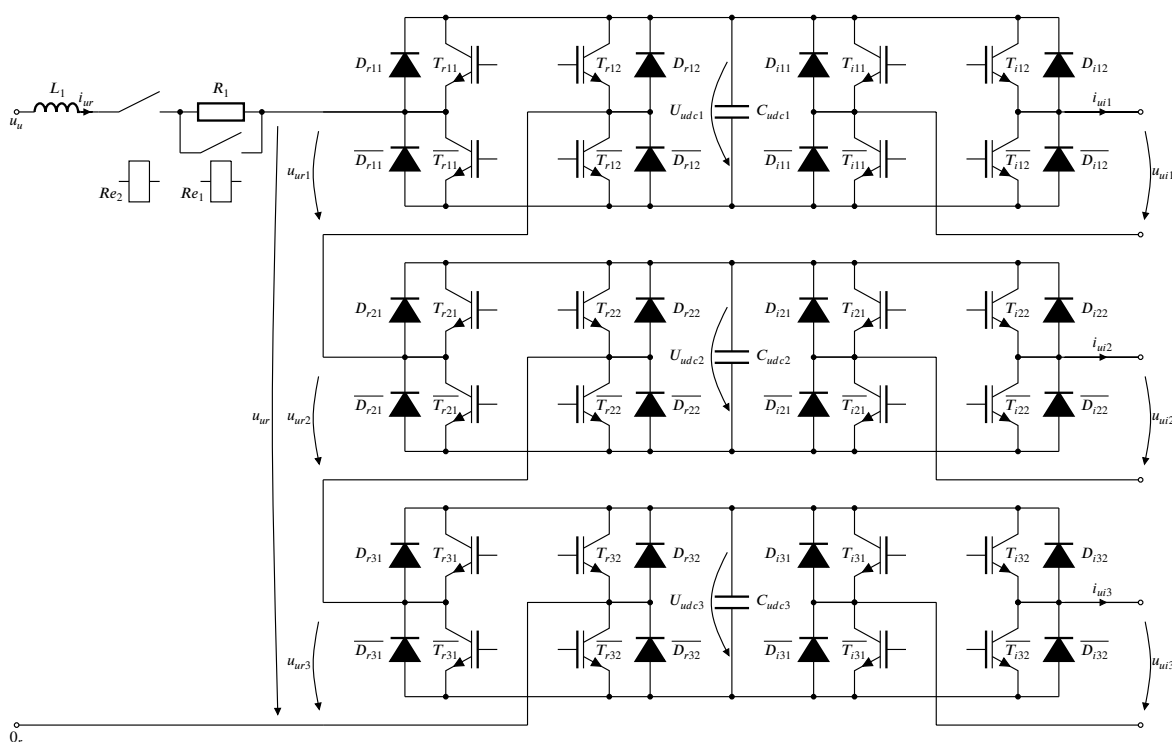
topologie je složena z dvojice H můstků, které sdílejí stejnosměrný meziobvod dle Obr. 4.7. Tato buňka je tříúrovňový měnič s výstupními hladinami napětí  $+U_{dc}$ ,  $0$ ,  $-U_{dc}$  (viz Tab. 4.3), kdy nulový vektor výstupního napětí lze dosáhnout dvěma kombinacemi sepnutí výkonových prvků. Oba dva H-můstky v této buňce jsou na sobě funkčně nezávislé, lze je pak provozovat v režimu usměrňovač a střídač. Sériově spojené H-můstky umožňují tzv. asymetrický režim, kdy na buňkách není stejné napětí meziobvodu, tím lze např. dosáhnout devíti úrovní výstupního napětí se dvěma H-můstkem oproti symetrické variantě se čtyřmi, více viz [3]. V této práci není asymetrický režim použit.

Sériovým spojením těchto dílčích stavebních buněk dostáváme zapojení jedné fáze měniče (Obr. 4.8). Třífázová varianta by pak použila tuto strukturu nezávisle třikrát. Ve schématu jsou opět naznačeny komplementární dvojice jednotlivých prvků. Schéma je složeno ze tří buněk, tomu odpovídají indexy jednotlivých prvků.

#### 4.3.1. PWM modulátor pro 7L-CHB měnič v FPGA

Pro řízení sériově řazených H můstků byl navržen modulární pulzně šířkový modulátor, ten se vyznačuje několika vlastnostmi požadovanými pro řízení měniče s topologií H můstků a to:

- možnost synchronního běhu několika instancí toho modulátoru
- vnitřně generuje modulační signál s posuvem o  $180^\circ$ , pak pro řízení celého H můstku stačí jeden modulační signál



Obr. 4.8.: Schéma jedné fáze nepřímého měniče kmitočtu se sériově řazenými H můstky

- umožňuje fázově posunout pilu pro generátor pulzně šířkové modulace se zachováním synchronního běhu více modulátorů
- implementuje generátor mrtvých časů

Blokové schéma modulátoru ukazuje Obr. 4.9. Tato entita je schopná obsloužit jeden kompletní H můstek, tj. polovinu buňky z Obr. 4.7. V závislosti na počtu sériově zapojených můstků pak závisí počet těchto dílčích bloků v FPGA.

Navržený modulátor má maximálně šestnáctibitové rozlišení a generuje časově i amplitudově symetrickou pilu v rozsahu uživatelem nastavené periody  $N_p$  dle Obr 4.10.

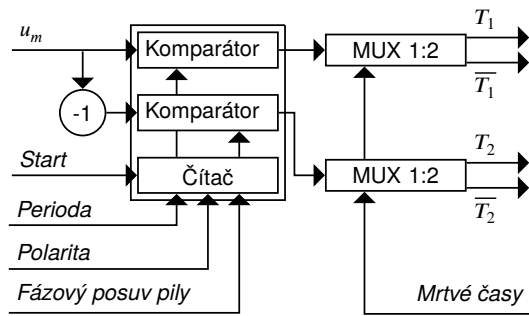
$$N_p = \frac{f_{clk}}{4 \cdot f_{PWM}} \quad (4.2)$$

kde

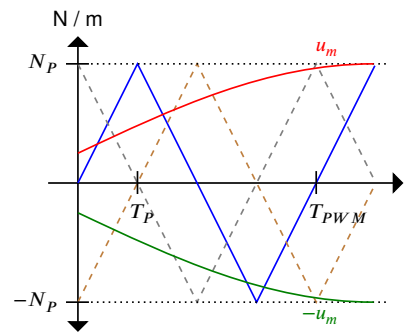
- $N_p$  ... číselná velikost periody pro PWM modulátor
- $f_{clk}$  ... frekvence hodin PWM modulátoru
- $f_{PWM}$  ... požadovaná frekvence PWM

Pomocí rovnice 4.2 lze vypočítat požadovanou hodnotu periody pro zadanou frekvenci PWM. V tomto konkrétním případě vyjde perioda modulátoru  $N_p$  rovna 31250, pro požadovanou výstupní frekvenci PWM pulzů  $f_{PWM}$  o velikosti 800 Hz při taktovacím kmitočtu modulátoru





Obr. 4.9.: Blokové schéma modulátoru pro jednu polovinu buňky měniče 7L-CHB



Obr. 4.10.: Komparační pila čítače CHB modulátoru

$f_{clk}$  rovném 100 MHz.

Modulační signál v rozsahu  $\pm 1$  nelze použít přímo pro řízení modulátoru, nejprve se musí vynásobit hodnotou periody. Entita modulátoru si vnitřně generuje invertovaný signál  $-u_m$ , tím je ušetřen jeden řídicí registr. Modulátor implementuje možnost fázově posunout svou komparační pilu v rozsahu  $\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$  (šedá, hnědá pila na Obr. 4.10). V kombinaci s možností měnit polaritu výstupních pulzů, je pokryt rozsah fázového posunutí v celém rozsahu  $\langle -\pi, \pi \rangle$ .

#### 4.4. Provozování víceúrovňových měničů s omezeným počtem synchronních PWM

První experimenty s víceúrovňovými měniči byly realizovány s pomocí vývojového kitu, který nedisponoval dostatečným počtem synchronních PWM výstupů. Dostupný počet synchronních PWM byl limitován mikrokontrolérem TMS320F28335 na maximálních 12. Ze vzorce 2.1 lze vypočítat, že s takovýmto hardwarem, bez doplňujících zařízení lze uřídit maximálně tříúrovňový třífázový střídač nebo usměrňovač.

Požadavek však byl na řízení čtyřúrovňového měniče s plovoucími kondenzátory a tříúrovňového s aktivním upínáním, obě tyto topologie mají shodně po šesti spínacích prvcích na fázi. Analýzou kombinací sepnutí jednotlivých součástek bylo zjištěno, že počet PWM signálů pro řízení obou topologií měničů lze redukovat a využít pro jejich ovládání i stávající hardware původního číslicového regulátoru.

##### 4.4.1. Návrh nízkoúrovňového kontroléru pro měniče 4L-FLC a 3L-ANPC

Vzhledem k faktu, že prvky víceúrovňových měničů spínají komplementárně, je možné zmenšit požadovaný počet řídicích signálů až na polovinu, ovšem za cenu větší složitosti řídicích obvodů. Ty bylo možné implementovat pomocí logických hradel nebo se nabízelo použi-

tí programovatelných logických obvodů. CPLD jsou velmi flexibilní a lze jimi nahradit velký počet hradel, zároveň lze jejich funkci dodatečně upravit dle aktuálního požadavku. Vstupně/výstupní vývody CPLD nemají fixní funkci, tu definuje uživatel, vychází pak výhodněji návrh plošného spoje. Omezení CPLD obvodů je především v použitých napěťových úrovních, běžně dostupné typy používají úrovně LVTTTL logiky, existují však i varianty s napájením 5 V, které pak mají klasické TTL úrovně, a tedy vyšší odolnost proti rušení než LVTTTL. Kontrolér je umístěn v bezprostřední blízkosti výkonových prvků, musí být proto odolný proti rušení generovanému při provozu měniče. Z tohoto pohledu jsou výhodnější hradla, neboť jsou schopna bez dalších obvodů fungovat s 15 V úrovněmi řídicích signálů (CMOS technologie), což zvyšuje šumovou imunitu.

Po zhodnocení výše uvedeného byla zvolena realizace pomocí programovatelného logického obvodu XC9572 [67]. Tento obvod je zástupcem CPLD napájených napětím 5 V, vstupní i výstupní úrovně má tudíž v úrovních TTL logiky. Byl vyvinut univerzální nízkoúrovňový kontrolér s tímto obvodem pro šest výkonových IGBT prvků, ovládaných pomocí budičů založených na HCPL-316 posílených obvodem od výrobce IXYS. Flexibilita kontroléru daná použitím CPLD obvodu nám umožnila použít stejný kontrolér jak u tříúrovňového měniče s upínacími diodami (3L-NPC) tak u čtyřúrovňového s plovoucími kondenzátory (4L-FLC), a to beze změn na DPS, pouze úpravou firmware v CPLD. Podrobněji o těchto měničích pojednávají kapitoly 5.2 a 5.1.

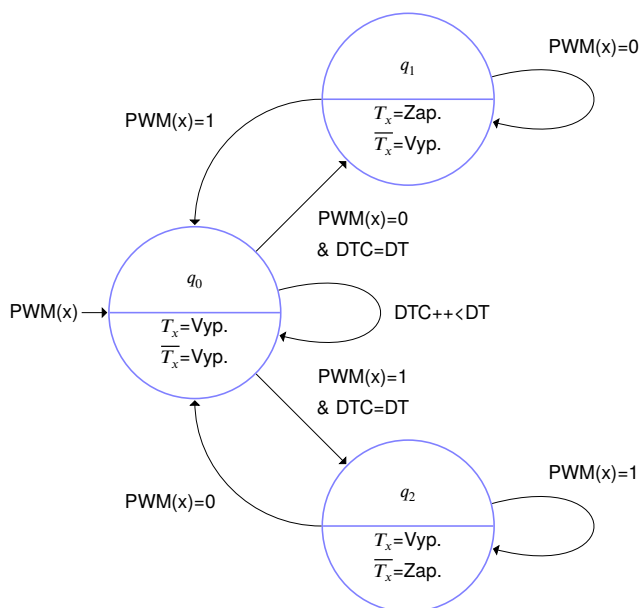
Kontrolér je fyzicky realizován jako modul pro šest výkonových prvků (tj. jedna fáze, viz Obr. A.7), k propojení s číslicovým regulátorem pak slouží jednoduchá DPS s příslušnými konektory, která zároveň propojuje signalizaci chyb jednotlivých modulů kontroléru mezi sebou a rozvádí napájecí napětí 24 V.

Logika v CPLD musí řešit rozdělení tří vstupních PWM signálů na šest výstupních PWM signálů přímo řídicích výkonové prvky. Signál nelze pouze rozbočit, bylo třeba implementovat generátor mrtvých časů. Ten je založen na principu stavového automatu, který má definovány tři stavy (viz Obr. 4.11):

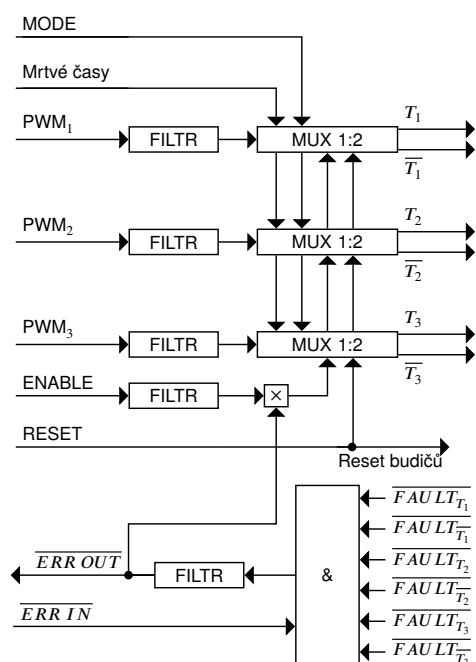
- Oba prvky vypnuty, stav  $q0$ .
- Zapnutý spodní prvek, stav  $q1$ .
- Zapnutý horní prvek, stav  $q2$ .

Tyto tři stavy postačují pro funkci kontroléru jako demultiplexeru jednobitového PWM signálu na dvoubitový. Mrtvé časy jsou aplikovány vždy při změně vstupního řídicího signálu, tj. v přechodu mezi stavy  $q0 \rightarrow q1$  nebo  $q0 \rightarrow q2$ .

Doplňková logika zajišťuje pomocí vstupů  $MODE$ ,  $ENABLE$ ,  $\overline{ERR\ IN}$ ,  $RESET$  a výstup  $\overline{ERR\ OUT}$  další funkcionalitu nutnou pro správnou funkci měniče, ta je shrnuta v Tab. 4.4. Blokové schéma firmware v CPLD je vidět na Obr. 4.12. Malá vzdálenost CPLD od výkonových



Obr. 4.11.: Funkční diagram multiplexeru 1:2 s generátorem mrtvých časů



Obr. 4.12.: Vnitřní blokové schéma demultiplexeru PWM signálu s generátorem mrtvých časů v CPLD

prvků si vynutila vložení digitálních filtrů na každý z PWM vstupů,  $ENABLE$  a  $\overline{FAULT}$  vstupy. Tyto filtry slouží k odstranění pulzů kratších než trojnásobek hodinového kmitočtu tohoto filtru. V principu jde o tříbitový posuvný registr, který spolu s komparátorem posuzuje, zda byla na vstupu filtru současná logická úroveň déle než tři periody hodin filtru. Pokud ano, aktuální hodnota na vstupu se přenese na výstup. Perioda hodin je  $800 \text{ ns}$ , takže filtr nepropustí pulzy kratší než  $2,4 \mu\text{s}$ .

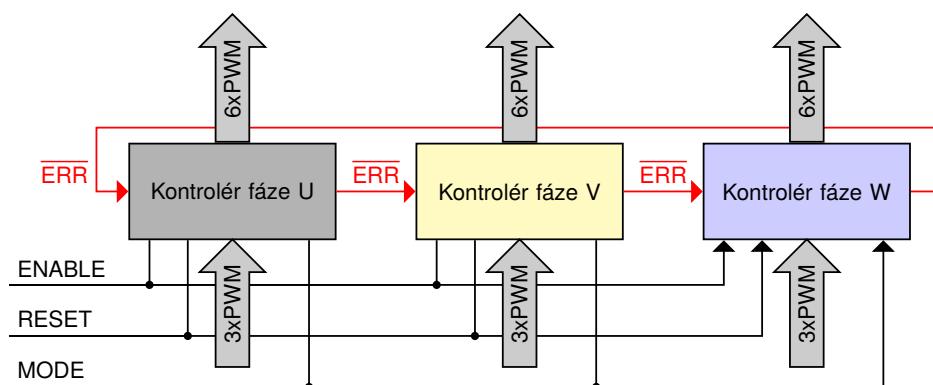
Každý měnič obsahuje tři tyto moduly, pro každou fázi jeden. Obr. 4.13 zobrazuje vzájemné propojení jednotlivých modulů. Z obrázku je patrné, že ovládací signály jsou sloučeny a signál  $\overline{ERR}$  je zacyklen přes všechny moduly tak, aby vybavení desaturační ochrany na kterémkoli prvku měniče způsobilo odstavení všech tří fází.

Ovládací signál  $MODE$  má specifické použití u obou topologií měničů. V případě 4L-FLC umožňuje přepnutí měniče do tzv. přednabíjecího módu, kdy využíváme strukturu měniče pro přednabití plovoucích kondenzátorů. Více viz kapitola 5.2.1. U 3L-ANPC měniče je tento ovládací signál použit pro programové přepínání režimu. Pokud je  $MODE$  v log. 0, měnič pracuje v režimu NPC, pokud je v log. 1 pak jsou aktivní i upínací tranzistory - ANPC režim.

Použitím CPLD pro implementaci logiky měniče bylo dosaženo toho, že bylo možno použít stejný modul pro odlišné topologie měničů (4L-FLC a 3L-NPC). Změnou programu v obvodu pak bylo dosaženo příslušné změny funkčnosti bez jakýchkoli úprav zapojení desky

Tab. 4.4.: Funkce pinů kontroléru

Signál	Funkce	Typ
<i>MODE</i>	Přepínání režimů kontroléru (viz text)	vstup
<i>ENABLE</i>	Povoluje pulzy na PWM výstupech	vstup
<i>RESET</i>	Resetuje HCPL 316 a odblokovává logiku v CPLD	vstup
<i>ERR IN</i>	Indikace selhání od vedlejší fáze	vstup
<i>ERR OUT</i>	Indikace selhání pro vedlejší fázi	výstup



Obr. 4.13.: Blokové schéma propojení a řídicích signálů tří fází měniče 4L-FLC/3L-ANPC

s plošným spojem. Zároveň modul umožnil redukovat počet požadovaných PWM signálů z původních 18 na 9 plus 3 další asynchronní signály pro ovládání logiky v CPLD. Doplnkové asynchronní signály již nemusí být synchronní s PWM výstupy, lze tedy pro jejich ovládání využít kterýkoli z uživatelsky kontrolovatelných vývodů mikrokontroléru.

## 4.5. Dílčí závěr

Byly navrženy a pomocí vyvinutého MLC interface implementovány PWM modulátory pro zkoumané typy vícehladinových měničů. Bylo ověřeno, že navržený číslicový regulátor MLC interface, respektive jeho část s FPGA umožňuje realizovat modulátory pro víceúrovňové měniče a odstranit tak z pohledu regulačních algoritmů rozdílnosti mezi topologiemi. Je možné mít unifikované rozhraní mezi modulátorem v FPGA a nadřazeným mikrokontrolérem. Byly navrženy modulátory pro:

- Čtyřúrovňový měnič s plovoucími kondenzátory.
- Tříúrovňový měnič s upínacími diodami
- Sedmiúrovňový měnič s H-můstky.

Všechny tyto modulátory jsou jednoduše škálovatelné tak, aby umožnily řízení měničů s téměř neomezeným počtem hladin.

Dále byla prozkoumána možnost řídit jednodušší (s menším počtem hladin) víceúrovňové topologie měničů přímo pomocí mikrokontroléru, který má jen omezený počet synchronních PWM výstupů za pomoci jednoduché logiky v CPLD. Využitím CPLD místo obvodů složených přímo z hradel bylo dosaženo jednoduššího návrhu desky plošného spoje pro jednotlivé bloky měniče a zároveň bylo možné tyto nízkoúrovňové řídicí desky využít pro více různých topologií měničů (3L-NPC, 3L-ANPC a 4L-FLC). Jednotlivá pracoviště s použitými topologiemi měničů jsou ukázána v Příloze B.

## 5. Řízení víceúrovňových měničů a komplexních měničových sestav

### 5.1. Tříúrovňový napěťový nepřímý měnič kmitočtu s upínacími diodami

#### 5.1.1. Regulace napěťového pulzního usměrňovače s topologií 3L-NPC

Start měniče je zajištěn pomocí dvojice relé  $Re_1$  a  $Re_2$ , na rozdíl od měniče 4L-FLC (viz 5.2.1) není nutné aktivně spínat žádné výkonové prvky. Pomocí  $Re_2$  se připojí měnič k síti a pak algoritmus pouze počká na dostatečné nabití kondenzátorů v meziobvodu. Jakmile je dosaženo napětí o velikosti 80 % amplitudy napětí sítě, sepne se  $Re_1$ , tím se přemostí přednabíjecí odpor  $R_1$ , spustí se regulační algoritmus NPU a měnič je připraven k činnosti.

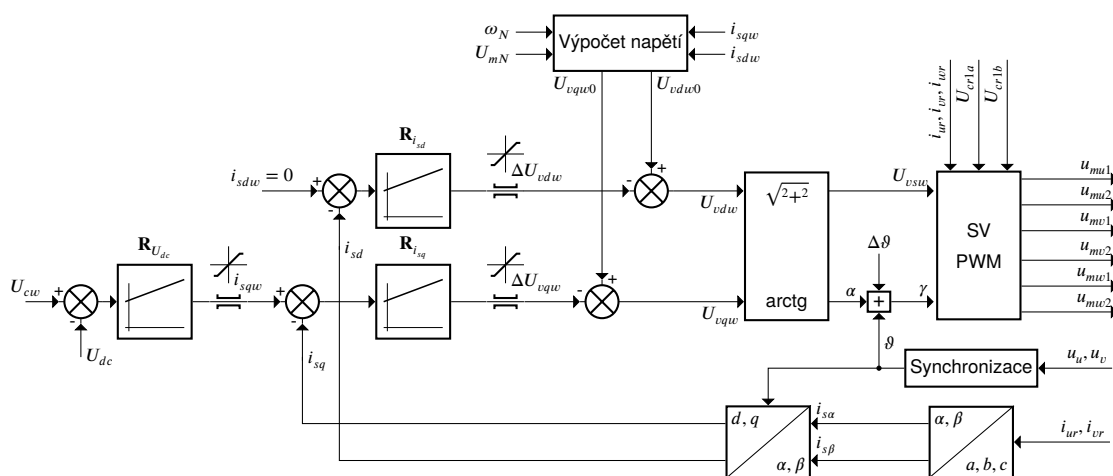
Algoritmus řízení napěťového pulzního usměrňovače je rozdělen na dvě části: generátor pulzně šířkové modulace je umístěn v FPGA, zatímco výpočet regulačního algoritmu, kombinaci sepnutí výkonových prvků spolu s měřeními a ochranami zajišťuje mikrokontrolér.

Základní schéma navržené regulace je vykresleno na Obr. 5.1. Jedná se o vektorovou regulaci v  $d$ - $q$  souřadnicích skládající se ze dvou smyček: regulace proudů  $i_{sq}$  a  $i_{sd}$ , přičemž je proud  $i_{sd}$  regulován na nulovou hodnotu, tím je zajištěn odběr sinusového proudu ze sítě s  $\cos(\varphi)$  blízkému jedné, a nadřazené regulace napětí v meziobvodu. Navržené algoritmy řízení vycházejí z práce [68].

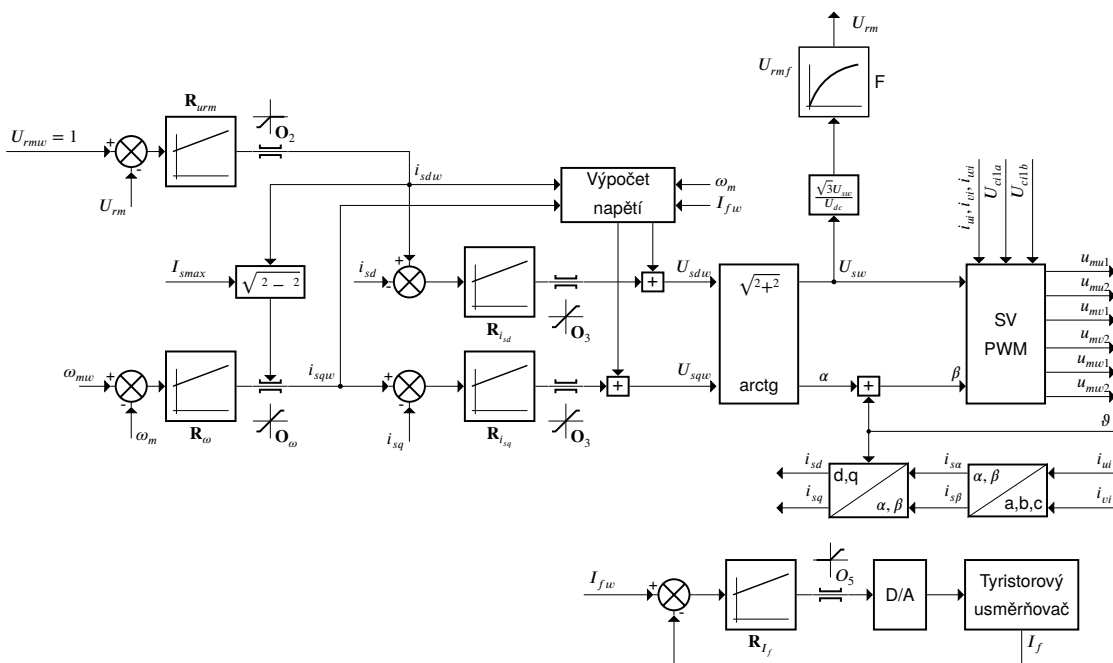
Řídící proměnné  $\gamma$  a  $U_{vsw}$  jsou dále zpracovány algoritmem SV PWM, který zajišťuje také balancování napětí na obou kondenzátorech (viz práce [69]) a výsledné modulační signály pro jednotlivé komplementární dvojice normované na velikost periody PWM modulátoru jsou odeslány do FPGA, které generuje příslušné průběhy pro řízení usměrňovače.

#### 5.1.2. Regulace synchronního motoru s buzením měničem topologie 3L-NPC

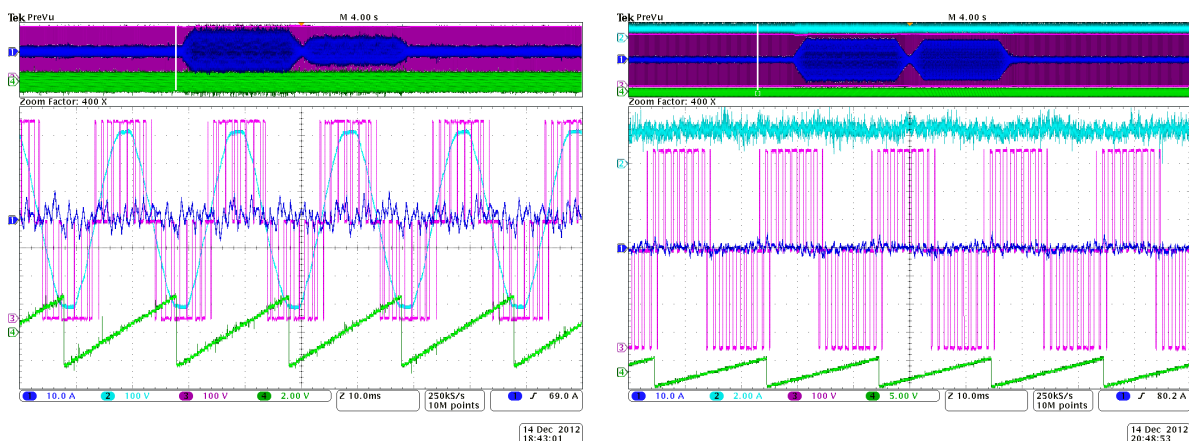
Ke střídačové části měniče byl připojen synchronní motor s budícím vinutím. Navržený regulační algoritmus vychází z práce [68], jeho schéma je na Obr. 5.2. Do regulační struktury přibyl PI regulátor budícího proudu  $I_f$ , který přes digitálně-analogový převodník dostupný na MLC interface ovládá externí řídicí jednotku třífázového tyristorového usměrňovače (viz práce [70]).



Obr. 5.1.: Blokové schéma regulace napěťového pulzního usměřovače pro 3L-NPC



Obr. 5.2.: Blokové schéma regulace synchronního motoru pro 3L-NPC



(a) Usměrňovač, CH1 (modrá): fázový proud usměrňovače  $i_{ur}$  [10 A/d], CH2 (světle modrá): fázové napětí sítě  $u_u$  [100 V/d], CH3 (fialová): fázové napětí měniče  $u_{ur}$  [100 V/d], CH4 (zelená): řídicí úhel  $\gamma$  [170° el./d]

(b) Střídač, CH1 (modrá): fázový proud střídače  $i_{ui}$  [10 A/d], CH2 (světle modrá): budící proud rotoru  $I_f$  [2 A/d], CH3 (fialová): fázové napětí měniče  $u_{ui}$  [100 V/d], CH4 (zelená): poloha rotoru  $\vartheta$  [360° el./d]

Obr. 5.3.: Průběhy nezatíženého měniče 3L-NPC,  $U_{dc}=700$  V, v ustáleném stavu

### 5.1.3. Experimentální ověření funkce rekuperačního měniče s upínacími diodami

Správná funkce navrženého regulačního algoritmu a modulátoru byla ověřena pomocí experimentů na postaveném laboratorním prototypu měniče o výkonu 35 kVA. K měniči byl připojen synchronní motor s vinutým rotorem (parametry viz Tab. C.1) a jako jeho mechanická zátěž byl použit skuplovaný asynchronní motor (parametry v Tab. C.2) napájený z 4L-FLC měniče.

Chod naprázdno zobrazují oscilogramy Obr. 5.3a pro usměrňovač a Obr. 5.3b pro střídač. Z obou oscilogramů je vidět správná funkce modulátoru.

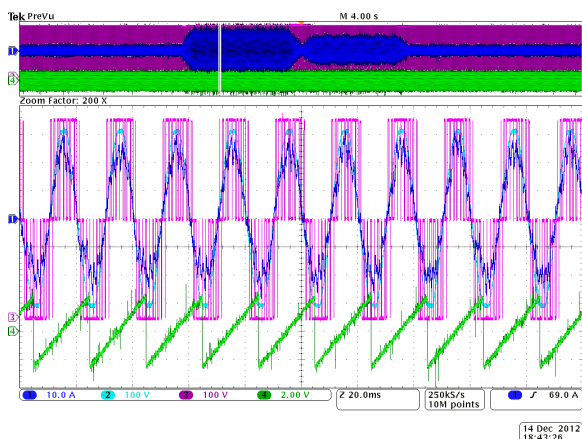
Na obrázcích Obr. 5.4a a 5.4b jsou průběhy napěťového pulzního usměrňovače v obou možných funkčních režimech při zátěži 10 kW. Pro střídač pak jsou výsledky měření obou funkčních režimů pro zatížení 10 kW na obrázcích Obr. 5.5a a 5.5b.

## 5.2. Čtyřúrovňový napěťový nepřímý měnič kmitočtu s plovoucími kondenzátory

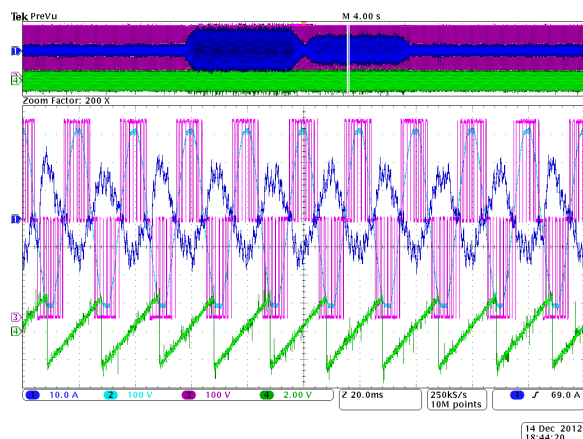
### 5.2.1. Přednabíjení víceúrovňových měničů s plovoucími kondenzátory

Start měniče s plovoucími kondenzátory je na rozdíl od dvouúrovňových měničů komplikovanější. Je potřeba vyřešit problém, kdy je nutné přednabít nejen kondenzátory v meziobvodu, ale také plovoucí kondenzátory. Ty však v běžném režimu měniče nelze připojit k meziobvodu s ohledem na komplementární spínání výkonových prvků. V článku [71] je rozebrána metoda přednabíjení jednofázového napěťového pulzního usměrňovače, která je dobře pou-



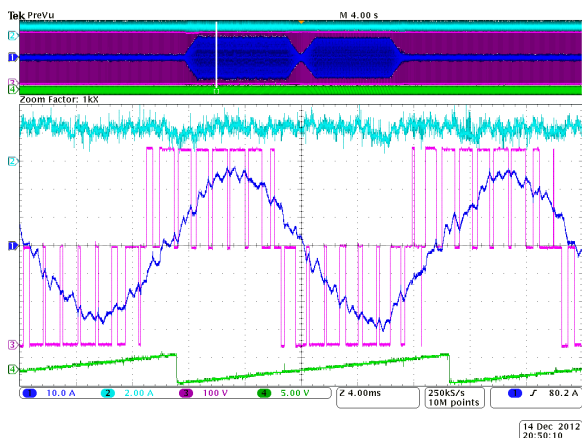


(a) Usměrňovačový chod

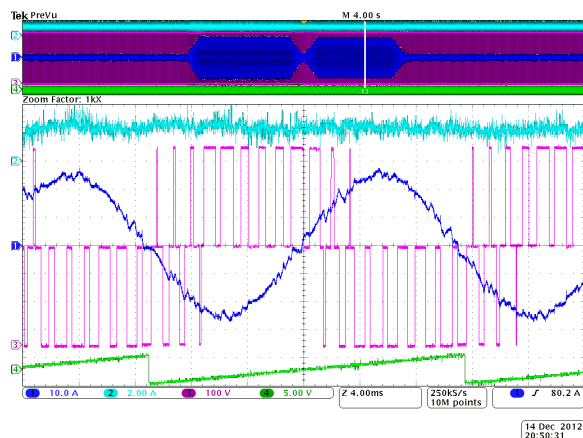


(b) Střídačový chod

Obr. 5.4.: Napěťový pulzní usměrňovač 3L-NPC při zátěži 10 kW,  $U_{dc}=700$  V, v ustáleném stavu, CH1 (modrá): fázový proud usměrňovače  $i_{ur}$  [10 A/d], CH2 (světle modrá): fázové napětí sítě  $u_u$  [100 V/d], CH3 (fialová): fázové napětí měniče  $u_{ur}$ [100 V/d], CH4 (zelená): řídicí úhel  $\gamma$  [170° el./d]

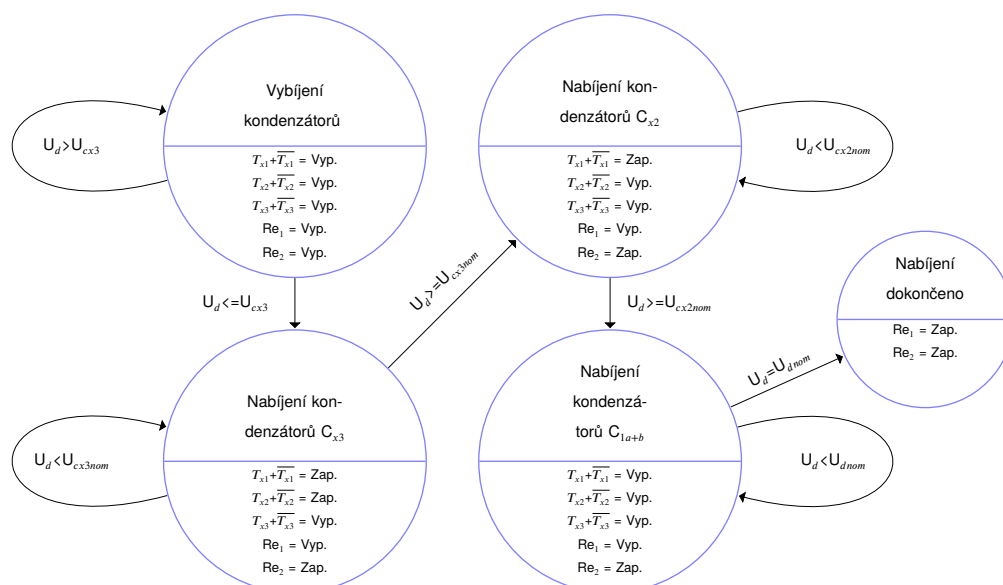


(a) Usměrňovačový chod



(b) Střídačový chod

Obr. 5.5.: Střídač 3L-NPC při zatížení 10 kW,  $U_{dc}=700$  V, v ustáleném stavu CH1 (modrá): fázový proud střídače  $i_{ui}$  [10 A/d], CH2 (světle modrá): budící proud rotoru  $I_f$  [2 A/d], CH3 (fialová): fázové napětí měniče  $u_{ui}$  [100 V/d], CH4 (zelená): poloha rotoru  $\vartheta$  [360° el./d]



Obr. 5.6.: Vývojový diagram přednabíjecího cyklu měniče 4L-FLC

žitelná pro NPU, avšak u střídače není použitelná. Metoda byla tedy upravena tak, aby byla univerzální, tj. použitelná i pro střídačovou část nepřímého měniče kmitočtu.

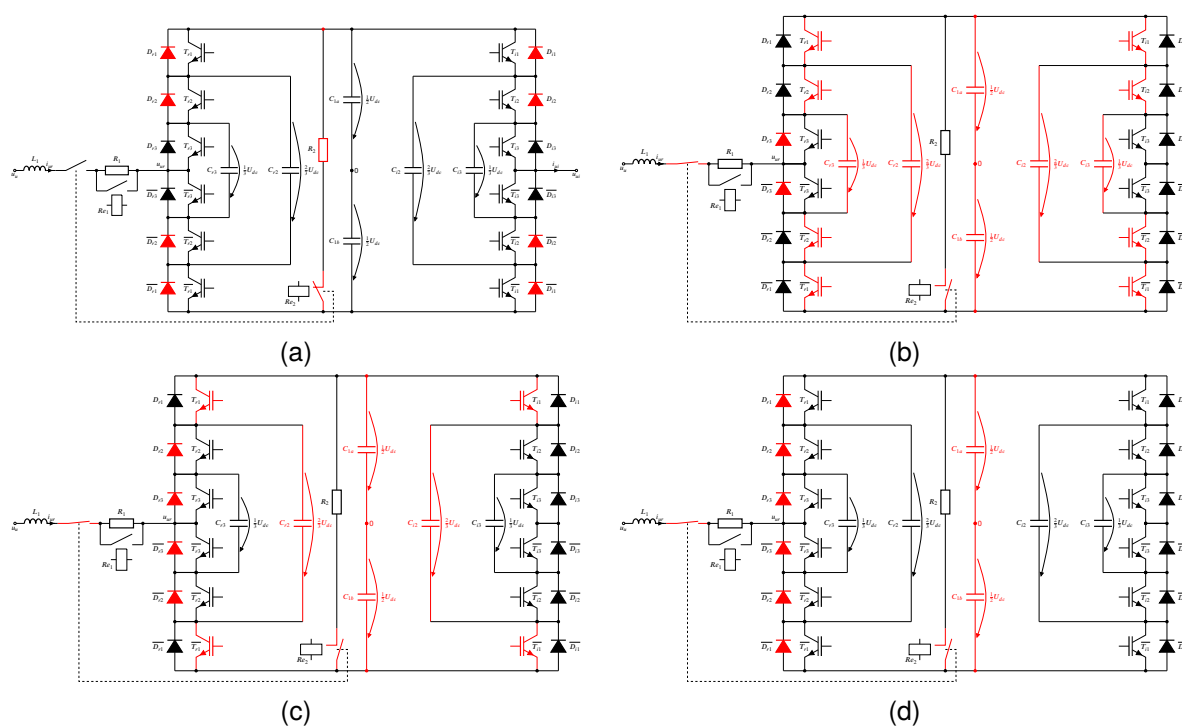
Každá fáze 4L-FLC má implementován nízkourovňový kontrolér popsany v kapitole 4.4.1, pomocí řídicího vstupu *MODE* lze přepínat mezi komplementárním režimem a přednabíjecím režimem. Přepnutí do přednabíjecího režimu deaktivuje tranzistory  $T_{x3}$  a  $\overline{T}_{x3}$  a dále změní ovládání zbylých dvou párů spínacích prvků tak, že log. 1 na PWM vstupu sepne oba prvky najednou. Pak je možné realizovat algoritmus přednabíjení vycházející z vývojového diagramu na Obr. 5.6. Tento platí konkrétně pro 4L-FLC, ale obecně je možné tento postup aplikovat na měnič s jakýmkoli počtem úrovní. Tím, že jsou spínány oba prvky z komplementární dvojice není potřeba rozlišovat, zda se přednabíjí střídač či usměrňovač.

Podmínkou správné funkce algoritmu je, aby napětí na plovoucích kondenzátorech v okamžiku přechodu do přednabíjecího režimu splňovalo podmínku:

$$U_{C_{i3}} \geq U_{C_{i2}} \geq U_{C_{1a+b}} \quad \cup \quad U_{C_{r3}} \geq U_{C_{r2}} \geq U_{C_{1a+b}} \quad (5.1)$$

Podmínku lze splnit cíleným vybíjením meziobvodu. Vyrovnání jednotlivých napětí na kondenzátorech zajistí zpětné diody u IGBT. Situaci osvětluje obrázek Obr. 5.7a, měnič je odpojen od sítě, kondenzátory se vybíjejí pomocí neseprnutého relé  $Re_2$ ,  $R_2$  a osmice diod (červeně zvýrazněné).

Při přednabíjení je měnič (Obr. 5.7b) připojen k síti přes nabíjecí odpor  $R_1$  seprnutím  $Re_1$ , což zároveň odpojí vybíjecí odpor  $R_2$  v meziobvodu. Následně seprnutím čtveřice tranzistorů v jedné větvi jsou spojeny všechny kondenzátory paralelně, diody  $D_{r3}$  a  $\overline{D}_{r3}$  pak spolu se zbylými dvěma fázemi (pro přehlednost zde nejsou zobrazeny) měniče tvoří šestipulzní diodový usměrňovač, přes který se nabíjí meziobvod a zároveň plovoucí kondenzátory.



Obr. 5.7.: Kroky přednabíjení měniče 4L-FLC

Po dosažení nominálního napětí pro kondenzátory  $C_{r3}$  a  $C_{i3}$ , tj.  $\frac{1}{3} U_{dc}$ , jsou odpojeny od meziobvodu vypnutím tranzistorů s indexem 2 (Obr. 5.7c). Nyní se sleduje nabíjení kondenzátorů  $C_{r2}$  a  $C_{i2}$ . Jakmile dosáhnou napětí  $\frac{2}{3}$  nominálního  $U_{dc}$ , vypnou se tranzistory s indexem 1. V posledním kroku (Obr. 5.7d) nabíjíme již jen kondenzátory meziobvodu. Po dosažení nominálního napětí v meziobvodu přepneme nízkourovňový kontrolér do komplementárního režimu a pustíme řídicí algoritmy usměrňovače a měniče.

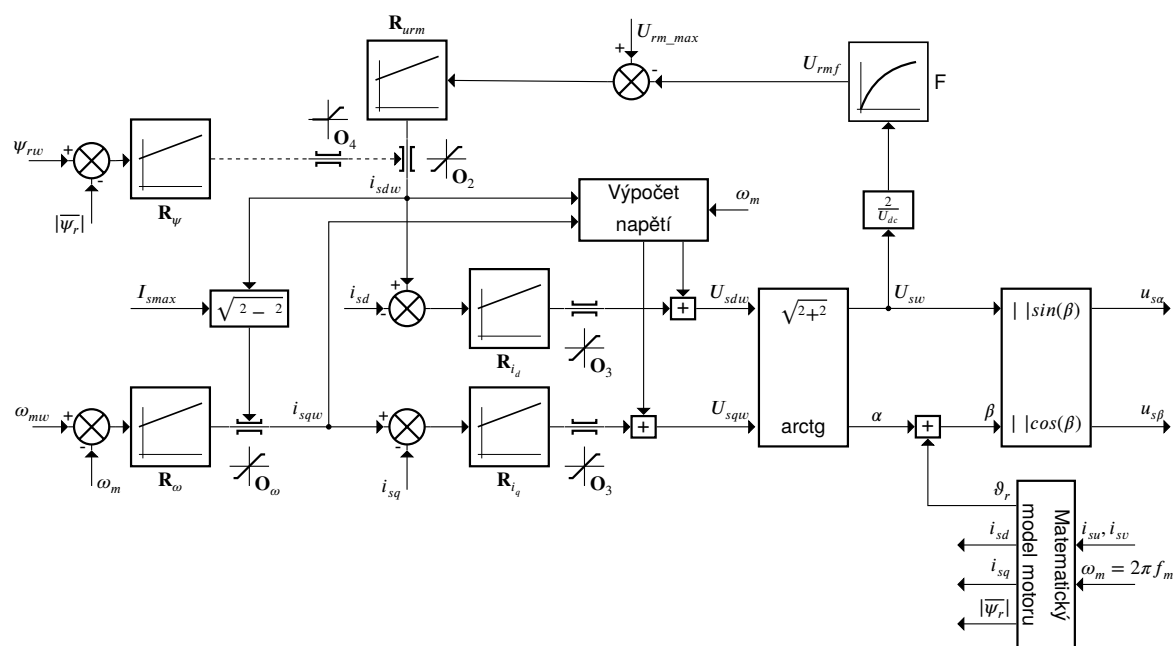
### 5.2.2. Regulace napěťového pulzního usměrňovače s topologií 4L-FLC

Navržený algoritmus řízení napěťového pulzního usměrňovače je rozdělen na dvě závislé části, kdy je modulátor pulzně šířkové modulační spolu s vyvažováním napětí na kondenzátorech implementován v FPGA, zatímco výpočet regulačních algoritmů spolu s měřeními a ochranami zajišťuje mikrokontrolér.

Základní schéma navržené regulace je naznačeno na Obr. 5.8. Jedná se o vektorovou regulaci v  $d-q$  souřadnicích. Algoritmy řízení vycházejí (stejně jako u 3L-NPC) z práce [68].

Výsledné modulační signály jsou dále přepočteny do souřadného systému  $\alpha, \beta$ , tím není třeba počítat funkce sinus a kosinus v FPGA, jelikož je implementace těchto algoritmů buď náročná na zdroje (použijeme-li ROM tabulku) nebo na dobu výpočtu v případě použití algoritmu CORDIC.



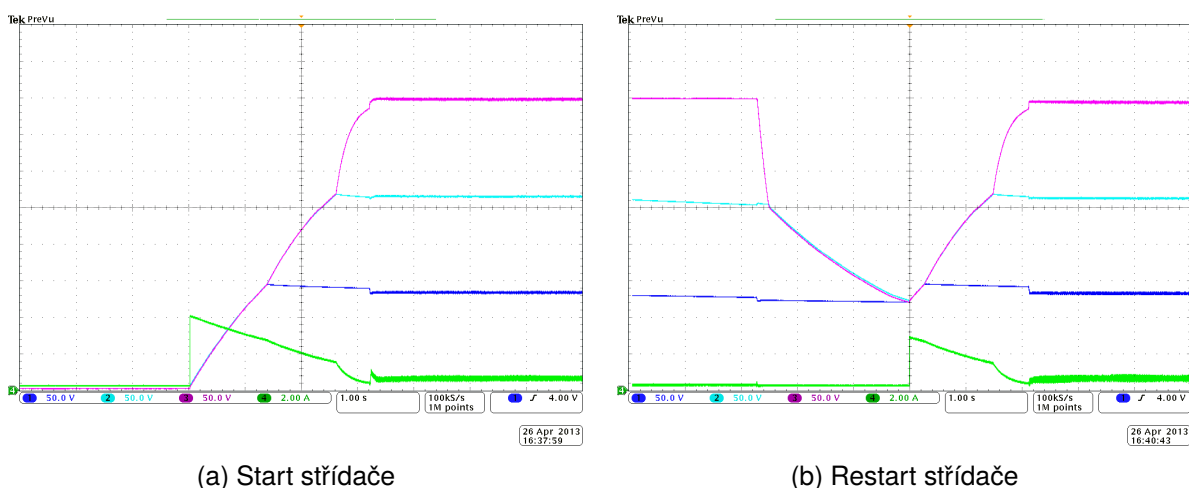


Obr. 5.9.: Regulační schéma pohonu s asynchronním motorem pro 4L-FLC

#### 5.2.4. Experimentální ověření přednabíjecí procedury pro 4L-FLC

Přednabíjecí algoritmus byl nejprve otestován na 4L-FLC střídači, který měl stejnosměrný meziobvod napájený z dynama přes přednabíjecí odpor. Výsledné průběhy pak zobrazují Obr. 5.10a a 5.10b. V prvním případě je vidět postupné nabíjení plovoucích kondenzátorů střídače. Z oscilogramu je patrné, že jsou plovoucí kondenzátory mírně přebíjeny a pak po startu střídače se jejich napětí dorovná na správnou hodnotu. Toto chování kompenzuje vybíjení plovoucích kondenzátorů vlivem připojeného vybíjecího odporu, když jsou odpojeny a nabíjí se kondenzátory s vyšší pracovní hodnotou napětí. Druhý oscilogram zobrazuje restart měniče, kdy nejdříve vybijeme kondenzátory na napětí dané podmínkou (5.1). Tímto postupem lze ušetřit čas i energii v porovnání s úplným vybíjením meziobvodu na téměř nulovou hodnotu.

Po ověření správné funkce přednabíjení u střídače bylo implementováno kompletní řízení přednabíjecího cyklu z MLC interface pro celý měnič. S výhodou bylo využito PWR výstupů dostupných na interface, které umožňují přímo ovládat použítá výkonová relé s nominálním ovládacím napětím  $24\text{ V} = (Re_1 \text{ a } Re_2)$ . Měnič lze tedy ovládat kompletně z algoritmu implementovaného v mikrokontroléru případně i z FPGA. Obr. 5.11a ukazuje start měniče včetně přednabíjecí procedury.



Obr. 5.10.: Přednabíjení střídače 4L-FLC,  $U_{dc}=700\text{ V}$ , CH1 (modrá): napětí na kondenzátorech  $C_{i3}$  [50 V/d], CH2 (světle modrá): napětí na kondenzátorech  $C_{i2}$  [50 V/d], CH3 (fialová): napětí v meziobvodu  $U_{dc}$  [50 V/d], CH4 (zelená): proud v meziobvodu  $I_{dc}$  [2 A/d]

### 5.2.5. Experimentální ověření funkce rekuperačního měniče s plovoucími kondenzátory

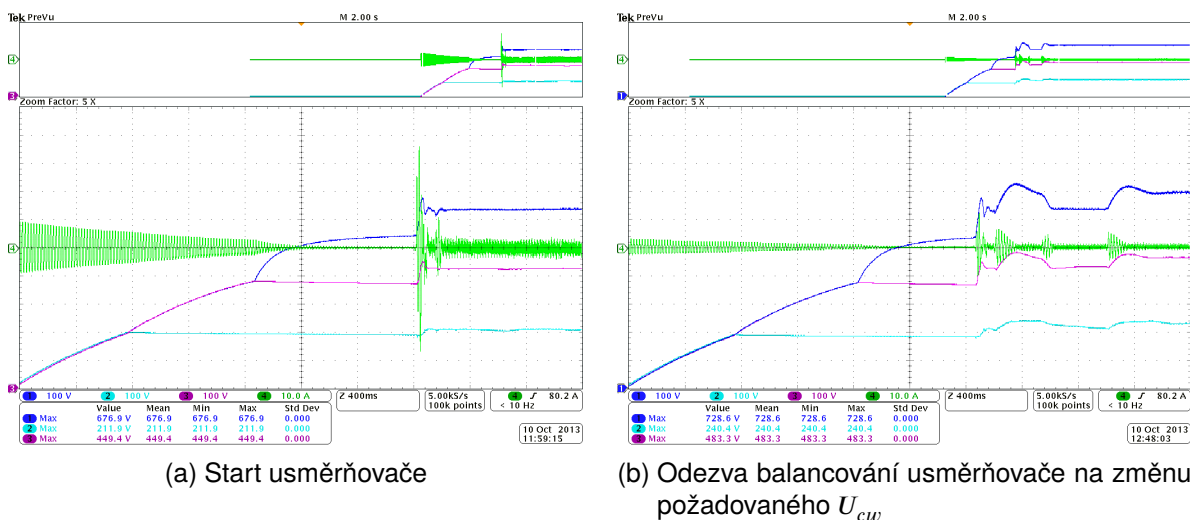
Experimentální ověření funkce prototypu 4L-FLC měniče bylo provedeno na testovacím soustrojí, které je složeno ze synchronního stroje s vinutou kotvou napájeného nepřímým měničem kmitočtu s měničem postaveném na topologii 3L-NPC a asynchronního motoru napájeného testovaným čtyřúrovňovým měničem. Parametry motorů viz příloha C. Oba měniče byly napájeny z jednoho přívodu 3x400 V.

Navržený hardware, modulátor v FPGA a řídicí algoritmy byly otestovány v základních pracovních režimech nepřímého měniče kmitočtu. Start měniče ukazují Obr. 5.11a a 5.11b. Na Obr. 5.11b je navíc vidět reakce řízení napěťového pulzního usměrňovače na změnu požadavku velikosti napětí v meziobvodu. Z oscilogramu je patrné, že navržená regulace napětí na plovoucích kondenzátorech pulzního usměrňovače je schopna si poradit i s dynamickými změnami napětí v meziobvodu.

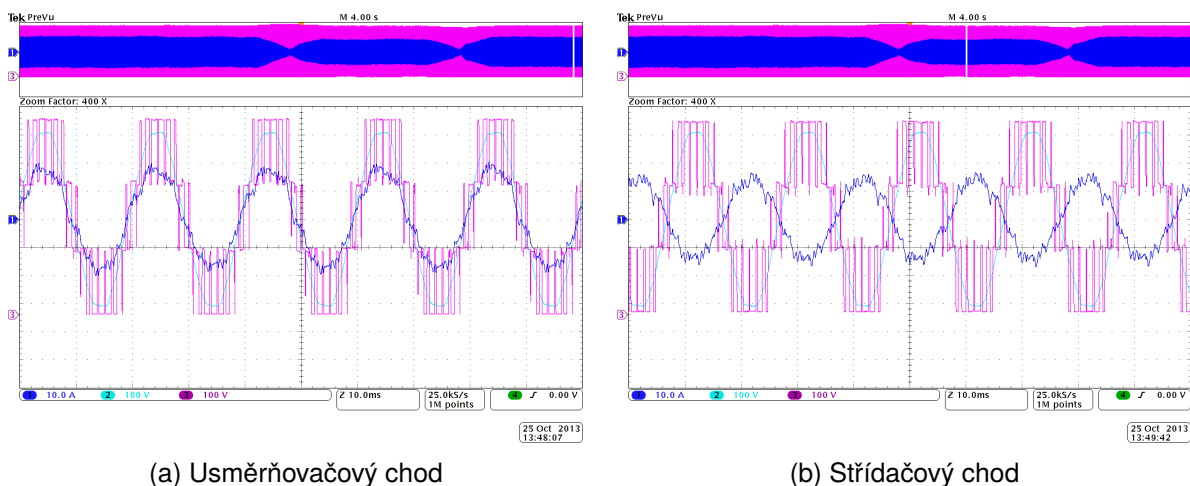
Následující oscilogramy Obr. 5.12a a 5.12b zobrazují průběhy proudu a napětí v ustáleném stavu při zatížení 9 kW v obou funkčních režimech napěťového pulzního usměrňovače.

Oscilogramy na Obr. 5.13a a 5.13b ukazují naměřené průběhy na střídači v ustáleném stavu při zatížení 9 kW a mechanické rychlosti 314,15  $\text{rad}\cdot\text{s}^{-1}$ .

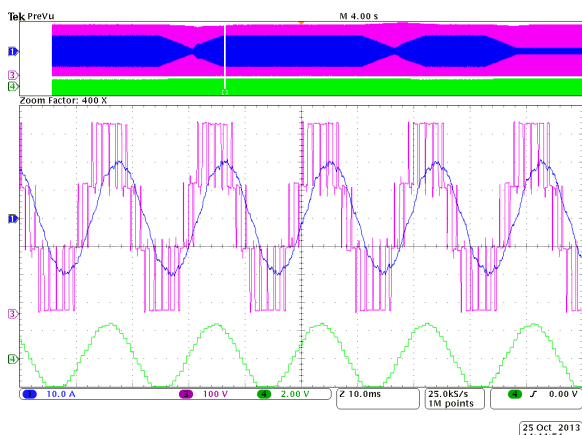
Reverzace otáčení nezatíženého pohonu ukazují pro usměrňovač Obr. 5.14a a pro střídač 5.14b. Změna otáčení byla z 94  $\text{rad}\cdot\text{s}^{-1}$  na -94  $\text{rad}\cdot\text{s}^{-1}$ . Z průběhů je patrné, že měnič a jeho balancovací algoritmus si je schopen bez problémů poradit i s dynamickou změnou zatížení.



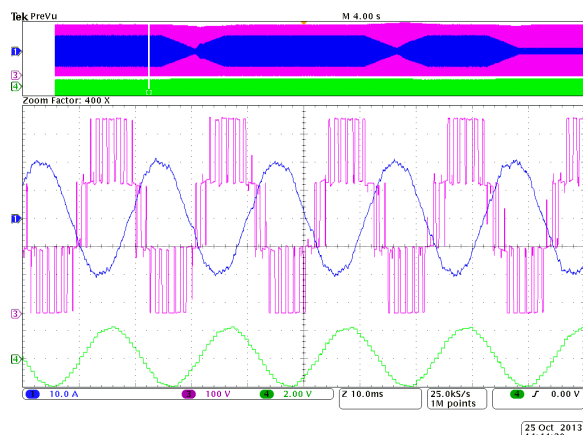
Obr. 5.11.: Usměrňovač 4L-FLC,  $U_{dc}=700$  V v ustáleném stavu, CH1 (modrá): napětí v meziobvodu  $U_{dc}$  [100 V/d], CH2 (světle modrá): napětí kondenzátoru  $C_{r3}$  [100 V/d], CH3 (fialová): napětí kondenzátoru  $C_{r2}$  [100 V/d], CH4 (zelená): fázový proud usměrňovače  $i_{ur}$  [10 A/d]



Obr. 5.12.: Usměrňovač 4L-FLC při zátěži 9 kW,  $U_{dc}=700$  V v ustáleném stavu, CH1 (modrá): fázový proud usměrňovače  $i_{ur}$  [10 A/d], CH2 (světle modrá): fázové napětí sítě  $u_u$  [100 V/d], CH3 (fialová): fázové napětí usměrňovače  $u_{ur}$  [100 V/d]

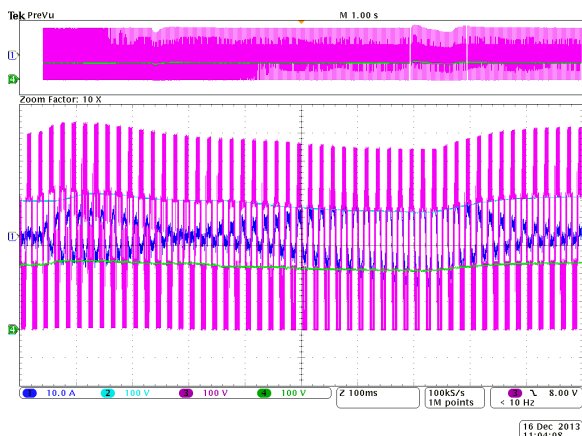


(a) Střídačový chod

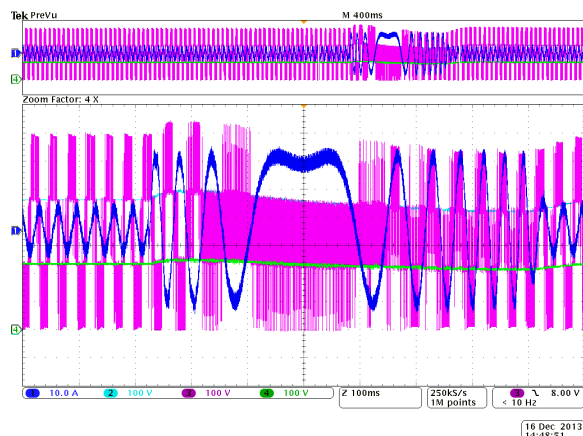


(b) Usměrňovačový chod

Obr. 5.13.: Střídač 4L-FLC při zatížení 9 kW,  $\omega_m = 314.15 \text{ rad.s}^{-1}$ ,  $U_{dc} = 700 \text{ V}$  v ustáleném stavu, CH1 (modrá): fázový proud střídače  $i_{ui}$  [10 A/d], CH3 (fialová): fázové napětí střídače  $u_{ui}$  [100 V/d], CH4 (zelená): požadované poměrné napětí pro modulátor [45 %/d]



(a) Průběhy pro usměrňovač



(b) Průběhy pro střídač

Obr. 5.14.: Usměrňovač 4L-FLC s nezatíženým motorem,  $U_{dc} = 700 \text{ V}$ , změna mechanické rychlosti z 94 na  $-94 \text{ rad.s}^{-1}$ , rampa  $250 \text{ rad.s}^{-1}$  el., CH1 (modrá): fázový proud měniče [10 A/d], CH2 (světle modrá): napětí na plovoucím kondenzátoru  $C_{r2}$  [100 V/d], CH3 (fialová): fázové napětí měniče [100 V/d], CH4 (zelená): napětí na plovoucím kondenzátoru  $C_{r3}$  [100 V/d]



### 5.3. Jednofázový sedmiúrovňový napěťový usměrňovač složený z H-můstků

#### 5.3.1. Jednofázová synchronizace pro napěťový pulzní usměrňovač

Řízení pulzního usměrňovače vyžaduje mít informaci o kmitočtu sítě a je nutné, aby bylo se sítí synchronizováno. Tuto funkcionalitu zajišťuje např. synchronizace založená na algoritmu SOGI PLL [73]. Regulační schéma SOGI PLL ukazuje Obr. 5.15. Signál pro fázový závěs nejprve prochází proporcionálním regulátorem ( $\mathbf{R}_{u\alpha}$ ) se zesílením  $K_{SOGI}$ , které ovlivňuje rychlost zavěšení, avšak čím vyšší je jeho hodnota, tím více si do regulační struktury zavedeme šum, offset a rušení. Odolnost proti rušení naopak zvyšují dva integrační bloky, které jsou zařazeny v cestě signálů  $u_\alpha$  a  $u_\beta$ . Tyto signály se pomocí rovnic Parkovy transformace (5.3) a (5.4) (vycházejících z Obr. 5.16) přepočítávají do souřadného systému  $d, q$ .

$$u_q = u_\alpha \cdot \cos(\vartheta_{PLL}) + u_\beta \cdot \sin(\vartheta_{PLL}) \quad (5.3)$$

$$u_d = u_\beta \cdot \cos(\vartheta_{PLL}) + u_\alpha \cdot \sin(\vartheta_{PLL}) \quad (5.4)$$

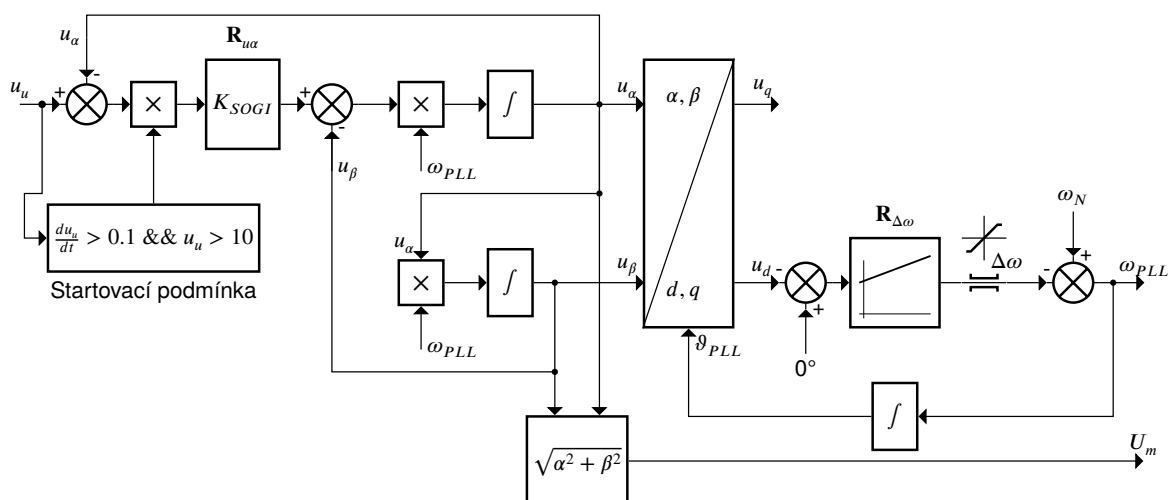
kde

$u_q, u_d$  ... napětí v rotujícím souřadném systému

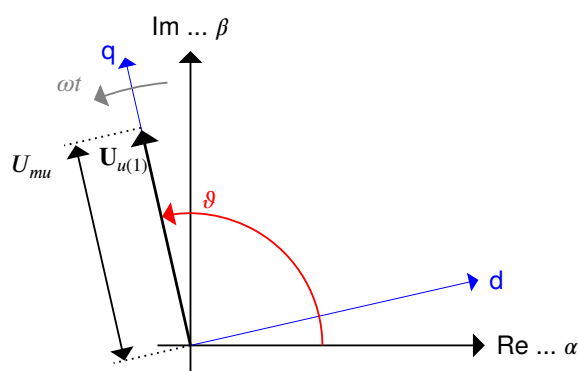
$u_\alpha, u_\beta$  ... napětí ve stojícím souřadném systému

$\vartheta_{PLL}$  ... poloha vektoru vstupního napětí z fázového závěsu

Souřadný systém  $d, q$  je orientován na vektor vstupního fázového napětí  $u_u$  (viz Obr. 5.16), takže navázané regulační bloky zajišťují regulaci napětí  $u_d$  na nulovou hodnotu. Toto zajišťuje PI regulátor ( $\mathbf{R}_{\Delta\omega}$ ), z něj získáváme přírůstek úhlové rychlosti  $\Delta\omega$ , kterou po



Obr. 5.15.: Blokové schéma SOGI PLL



Obr. 5.16.: Fázorový diagram vstupního napětí fázového závěsu SOGI

omezení odečteme od jmenovité úhlové rychlosti  $\omega_N$  a výsledkem je úhlová rychlost vstupního napájecího napětí. Integrací  $\omega_{PLL}$  získáme polohu vektoru napětí  $\vartheta_{PLL}$  a jeho velikost  $U_m$ , v souřadném systému  $\alpha - \beta$ .

Původní algoritmus byl upraven tak, aby se zlepšilo prvotní zavěšení po připojení k síti a zabránilo se případnému (i když málo pravděpodobnému) zavěšení do protifáze. Byl implementován detektor vrcholové hodnoty vstupního napětí, kdy je sledována derivace naměřeného fázového vstupního napětí. V okamžiku kdy je detekován rozdíl  $\Delta u_u$  větší než  $-0.1$  V při vzorkovací periodě  $50 \mu s$  a zároveň je fázové napětí kladné, odblokujeme algoritmus SOGI (startovací podmínka na Obr. 5.15). Rozdíl napětí  $\Delta u_u$ , který je třeba detekovat obecně vychází z (5.5). Po dosažení požadované startovací podmínky  $t = 0,005$  ms, která odpovídá úhlu  $90^\circ$  u sinusového průběhu a frekvenci 50 Hz dostáváme rovnici (5.6), zde již dosadíme pouze hodnotu periody vzorkování  $T_{vz}$  a efektivní velikost vstupního napětí  $U_m$ . Výsledkem je požadovaný rozdíl hodnot dvou po sobě následujících vzorků vstupního napětí, který použijeme do podmínky pro start SOGI PLL.

$$\Delta u_u = \sqrt{2} \cdot U_m \cdot (\cos(2 \cdot \pi \cdot f \cdot (t + T_{vz})) - \cos(2 \cdot \pi \cdot f \cdot t)) \quad (5.5)$$

$$\Delta u_u = \sqrt{2} \cdot U_m \cdot \cos\left(2 \cdot \pi \cdot \left(\frac{1}{4} + T_{vz}\right)\right) \quad (5.6)$$

kde

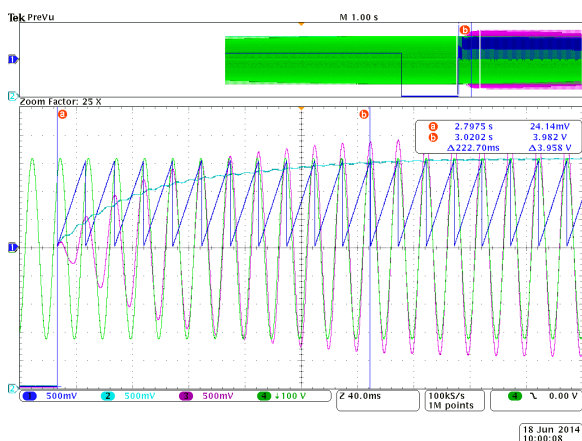
$\Delta u_u$  ... rozdíl vzorků vstupního napětí

$U_m$  ... amplituda vstupního napětí

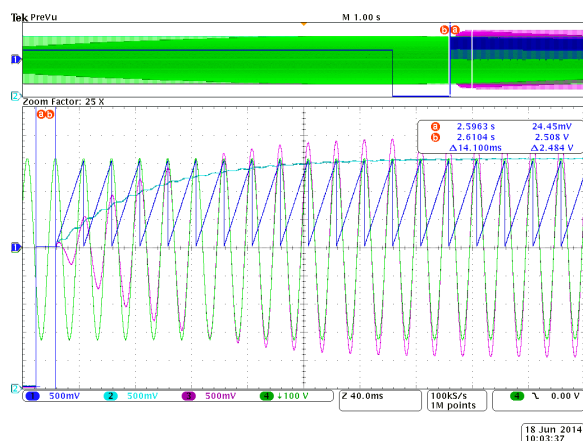
$f$  ... frekvence vstupního napětí

$T_{vz}$  ... perioda vzorkování

$t$  ... čas

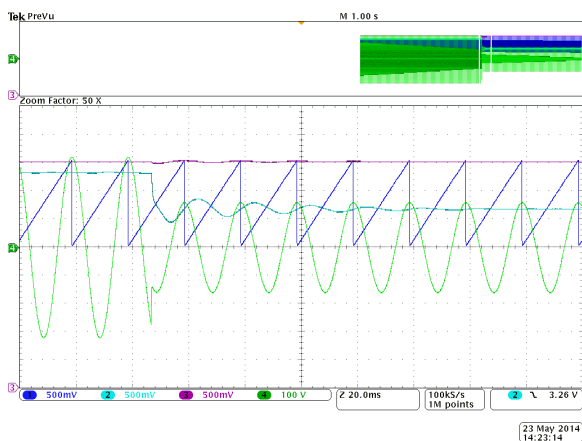
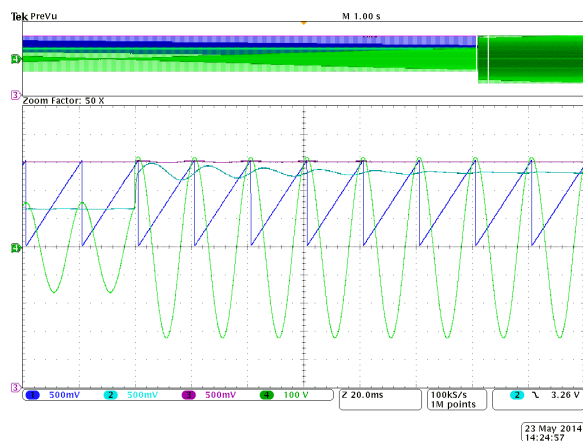


(a) Neoptimalizovaný start



(b) Optimalizovaný start

Obr. 5.17.: Start SOGI PLL, CH1 (modrá): poloha vektoru napětí sítě z PLL  $\vartheta_{PLL}$  [2 rad.s<sup>-1</sup>/d], CH2 (světle modrá): vyhodnocená amplituda napětí zdroje  $U_m$  [80 V/d], CH3 (fialová): (fialová): rekonstruované napětí sítě [75 V/d], CH4 (zelená): (zelená):  $u_u$  skutečné napětí napájecího zdroje [100 V/d]

(a) Skok napětí 230→115 V<sub>RMS</sub>(b) Skok napětí 115→230 V<sub>RMS</sub>

Obr. 5.18.: Odezva SOGI PLL na skokovou změnu vstupního napětí, CH1 (modrá): poloha vektoru napětí sítě z PLL  $\vartheta$  [2 rad.s<sup>-1</sup>/d], CH2 (světle modrá): vyhodnocená amplituda napětí zdroje  $U_m$  [80 V/d], CH3 (fialová): vyhodnocená  $\omega_{PLL}$  [105 rad.s<sup>-1</sup>/d], CH4 (zelená):  $u_u$  skutečné napětí napájecího zdroje [100 V/d]

Startovní podmínkou je zajištěno, že se SOGI PLL zavěsí správně a rychleji, než kdyby se nespouštěl v kontrolovaný okamžik. Na oscilogramech na Obr. 5.17 je patrný rozdíl v zavěšení fázového závěsu. Neoptimalizovaná verze se zasynchronizuje na síť za více než 200 ms, kdežto optimalizovaná bude mít zpoždění maximálně jednu periodu vstupního signálu, tj. 20 ms. Navíc tato startovací podmínka zajistí, že se SOGI PLL nezavěsí v protifázi vůči vstupnímu signálu.

### 5.3.1.1. Odezva synchronizace na bázi SOGI PLL na skokové změny parametrů sítě

Navržený algoritmus byl dále otestován na odolnost vůči skokovým změnám parametrů sítě, které mohou nastat. Odezva byla testována s pomocí laboratorního zdroje Agilent, generujícího základní signál  $u_u = 230 \cdot \sqrt{2} \cdot \sin(2 \cdot \pi \cdot 50 \cdot t)$ . K vykreslení sledovaných signálů  $\vartheta_{PLL}$ ,  $\omega_{PLL}$  a  $U_m$  byl použit D/A převodník, který je součástí MLC interface.

Obr. 5.18a zobrazuje odezvu SOGI PLL na skokovou změnu vstupního napětí z 230 na  $115 V_{RMS}$ , tj. o 50% směrem dolů. Obdobně Obr. 5.18b ukazuje skok napětí opačným směrem zpět na nominální hodnotu 230 V. Z oscilogramů je patrné, že skoky napětí nemají téměř žádný vliv na vyhodnocení vektoru napětí  $\vartheta_{PLL}$  ani na výslednou úhlovou rychlost  $\omega_{PLL}$ . Pouze vyhodnocovaná amplituda vstupního napětí  $U_m$  má mírné zákmity.

Mezi problematické změny parametrů sítě patří především změna její frekvence  $\omega$ . Změna frekvence je spíše pozvolného charakteru, její skoková změna je málo pravděpodobná, avšak provedené testy se zaměří výhradně na její skokovou změnu jakožto na extrémní stav. Provedené testy tedy skokově mění frekvenci v rozsahu 48 až 52 Hz viz dále. Odezva SOGI PLL byla měřena opět s využitím zdroje Agilent, který právě tyto skokové změny frekvence umožňuje a sledované výstupy ze SOGI PLL byly sledovány pomocí D/A převodníku jako v předchozím případě.

Výsledky shrnují obrázky Obr. 5.19a, 5.19b, 5.19c a 5.19d. Z oscilogramů plyne, že ustálení na nové hodnotě  $\omega_{PLL}$  a  $\vartheta_{PLL}$  není nijak rychlé, ale jak bylo zmíněno výše frekvence sítě se nemůže měnit skokem, obvyklá změna je v řádu desetin Hz za sekundu. Byly otestovány oba směry skoků, jak z 50 Hz nahoru na 52 Hz, tak i dolů na 48 Hz.

### 5.3.2. Regulační schéma pro jednofázový 7L-CHB měnič

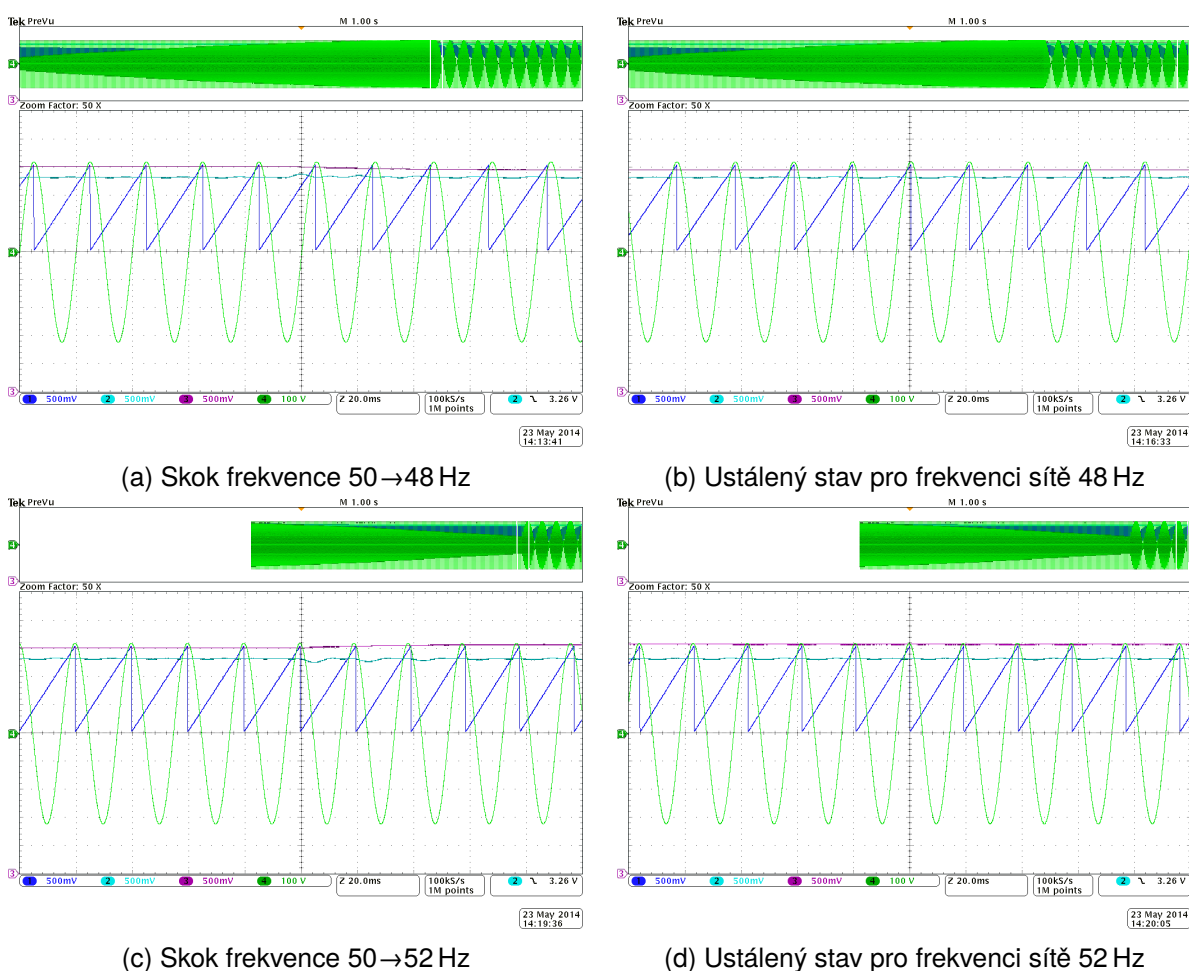
Kompletní navržené regulační schéma jednofázového NPU je na Obr. 5.20. Navržené řešení navazuje na práci [74]. Regulace využívá kombinace tří PI regulátorů pro řízení součtu napětí v meziobvodech jednotlivých H můstků a balancování napětí. Rezonanční regulátor spolu s modelem sítě regulují smyčku proudu  $I_m$  na střídavé straně pulzního usměrňovače.

Hlavní regulační smyčka má jako vstupní hodnotu sumu napětí na všech třech kondenzátorech, kterou porovnává s požadovaným napětím ve všech meziobvodech ( $U_{cw}$ ). Výsledná regulační odchylka pak po průchodu PI regulátorem  $R_{U_c}$  a omezovačem určuje požadovanou amplitudu proudu na střídavé straně měniče  $I_m$ . Ten je přepočten na kosinový průběh  $i_{uw}$ , který je porovnán s měřenou hodnotou fázového proudu měniče  $i_u$ , vzniklá odchylka vstupuje do rezonančního regulátoru  $R_{i_u}$ . Jeho výstup se pak sečte s výstupním napětím z modelu a s výstupem rezonančního regulátoru  $R_{3rd}$  kompenzujícího třetí harmonickou proudu měniče. Výsledek se následně normuje na sumu napětí kondenzátorů jednotlivých H můstků, tím je spočten absolutní modulační signál pro modulátor měniče. Ten se následně omezí na rozsah  $\pm 1$ , tato hodnota se posléze normuje na velikost periody PWM, ta se pak

zapiše do příslušného řídicího registru v FPGA a touto hodnotou je řízen první (master) H můstek. Omezovač na  $\pm 1$  zamezuje přetečení řídicí proměnné mimo rozsah  $int16\_t$  použitý pro normovanou hodnotu modulačního signálu v designu FPGA.

Balancování napětí na jednotlivých H můstcích používá metodu master-slave. Master je v tomto případě napětí na kondenzátoru  $U_{c1}$ ,  $U_{c2}$  a  $U_{c3}$  jsou s tímto napětím porovnávány. Regulátory balancování napětí na kondenzátorech  $R_{\Delta U_{c2}}$  a  $R_{\Delta U_{c3}}$  přímo ovlivňují amplitudu základního modulačního signálu  $u_{m1}$  tak, aby se napětí na všech třech kondenzátorech, resp. meziobvodech jednotlivých H můstků udržovalo stejné. Výsledné modulační signály se pak po omezení a znormování zapisují do řídicích registrů modulátoru v FPGA.

Výpočet modelu, pro první harmonickou, je založen na fázorovém diagramu dle Obr. 5.21. Ten vychází z náhradního obvodu na střídavé straně měniče, kdy pulzní usměr-



Obr. 5.19.: Odezva SOGI PLL na skokovou změnu frekvence, CH1 (modrá): poloha vektoru napětí sítě z PLL  $\vartheta$  [2 rad.s<sup>-1</sup>/d], CH2 (světle modrá): vyhodnocená amplituda napětí zdroje  $U_m$  [80 V/d], CH3 (fialová): vyhodnocená  $\omega_{PLL}$  [105 rad.s<sup>-1</sup>/d], CH4 (zelená):  $u_u$  skutečné napětí napájecího zdroje [100 V/d]



kde

$\vartheta_{HB}$  ... fázový posuv (přesazení) pily mezi jednotlivými můstky

$N_{HB}$  ... počet H-můstků na fázi

### 5.3.3. Experimentální ověření funkce napěťového pulzního usměrňovače s topologií 7L-CHB

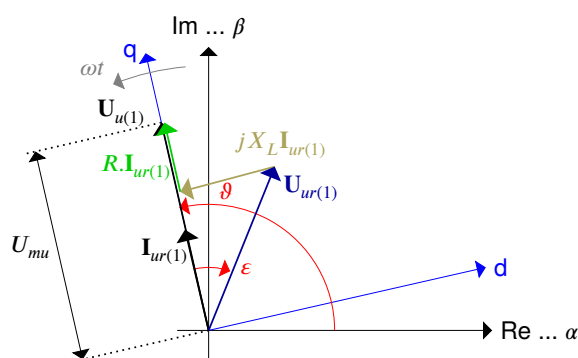
Pro ověření funkce byl použit jednofázový zdroj Agilent, stejně jako u SOGI PLL. Střídačová část topologie byla po připojení zatěžovacích výkonových odporů využita jako proměnlivá zátěž, která umožňovala i nesymetrické zatížení jednotlivých H-můstků. Pro regulaci zátěže byl realizován sinusový generátor s uživatelsky nastavitelnou výstupní frekvencí a amplitudou napětí.

Výsledky měření pak shrnují dále uvedené obrázky. První dvojice zobrazuje start usměrňovače (Obr. 5.22a) a následný ustálený stav (Obr. 5.22b) v případě, že nebyla připojena žádná zátěž. Na oscilogramech je vidět, že je navržené řízení schopno si poradit s dynamickou změnou napětí v meziobvodech jednotlivých můstků a udržet tak celkovou hodnotu na požadované velikosti.

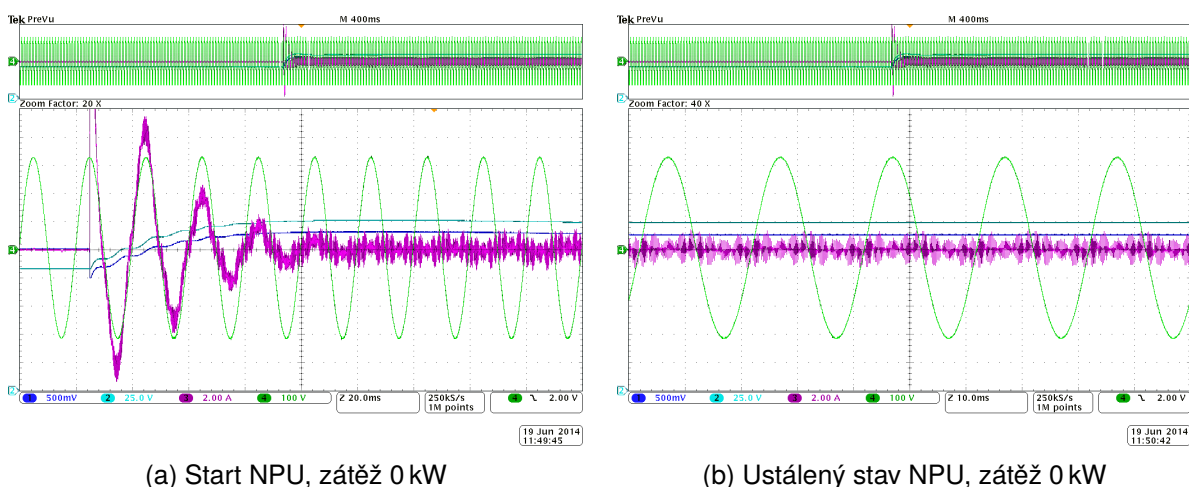
Navazují oscilogramy zobrazující výstupní průběhy z měniče. Obr. 5.23a ukazuje fázové výstupní napětí měniče a napětí na jednom z H-můstků v případě nezatíženého usměrňovače, Obr. 5.23b pak stejné průběhy pro zatížení 1 kW. Z oscilogramů je vidět, že se zatížením narůstá zvlnění napětí (druhou harmonickou síťového kmitočtu) v meziobvodu H-můstku.

Dále byl otestován vliv skoků napětí sítě na funkci usměrňovače při zátěži 1 kW, výsledky jsou zobrazeny na obrázcích Obr. 5.24a a 5.24b. Z těch je vidět korektní odezvu měniče na tyto rušivé vlivy.

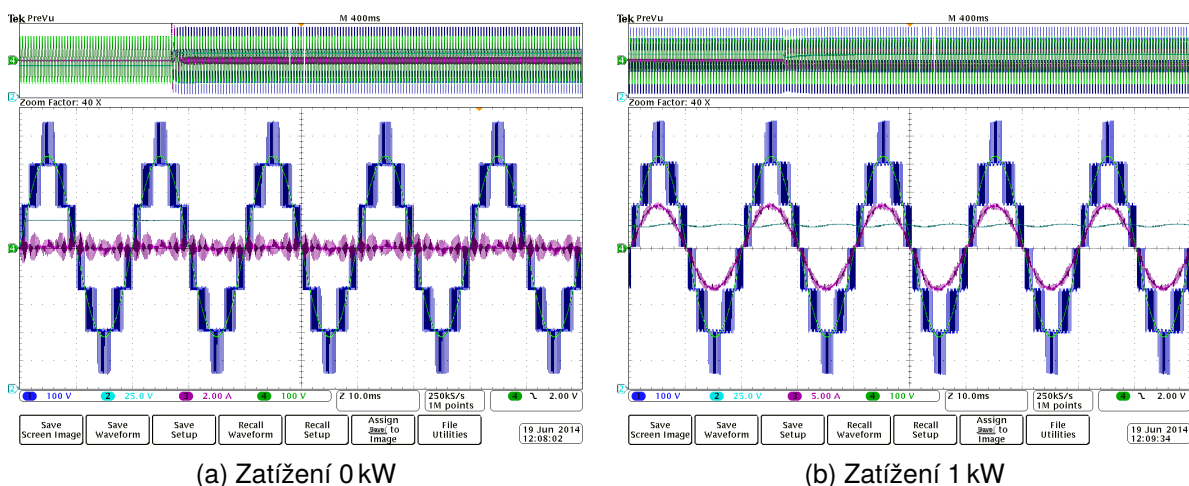
Druhý problematický parametr jednofázové sítě je frekvence, resp. její změna v čase a se zatížením sítě. Odolnost navrženého algoritmu řízení napěťového pulzního usměrňo-



Obr. 5.21.: Fázorový diagram vstupního napětí pulzního usměrňovače pro první harmonickou

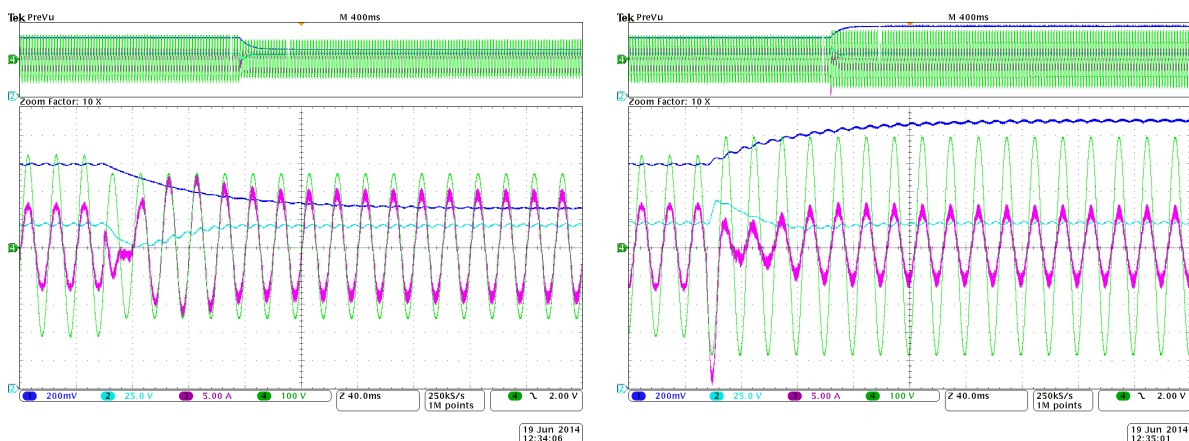


Obr. 5.22.: Start a ustálený stav napěťového pulzního usměrňovače 7L-CHB, CH1 (modrá): součet napětí v meziobvodech všech H můstků  $\sum U_{udcx}$  [80 V/d], CH2 (světle modrá): napětí meziobvodu můstku H1,  $U_{udc1}$  [25 V/d], CH3 (fialová): fázový proud usměrňovače  $i_{ur}$  [2 A/d], CH4 (zelená): fázové napětí sítě  $u_u$  [100 V/d]



Obr. 5.23.: Ustálené stavy 7L-CHB NPU,  $f=50$  Hz,  $U_{cw}=450$  V,  $U_{udc1,udc2,udc3}=150$  V, CH1 (modrá): fázové napětí měniče  $u_{ur}$  [100 V/d], CH2 (světle modrá): napětí meziobvodu můstku H1,  $U_{udc1}$  [25 V/d], CH3 (fialová): fázový proud usměrňovače  $i_{ur}$  [2 A/d]/[5 A/d], CH4 (zelená): fázové napětí sítě  $u_u$  [100 V/d]





(a) Skok napětí sítě  $230 V_{rms} \rightarrow 184 V_{rms}$ , zátěž 1 kW

(b) Skok napětí sítě  $230 V_{rms} \rightarrow 276 V_{rms}$ , zátěž 1 kW

Obr. 5.24.: Vliv skoků fázového napětí sítě na 7L-CHB,  $U_{cw}=450 V$ ,  $U_{udc1,udc2,udc3}=150 V$ ,  $f=50 Hz$ , CH1 (modrá): fázové napětí ze  $U_m$  SOGI PLL [40.65 V/d], CH2 (světle modrá): napětí meziobvodu můstku H1,  $U_{udc1}$  [25 V/d], CH3 (fialová): fázový proud usměrňovače  $i_{ur}$  [5 A/d], CH4 (zelená): fázové napětí sítě  $u_u$  [100 V/d]

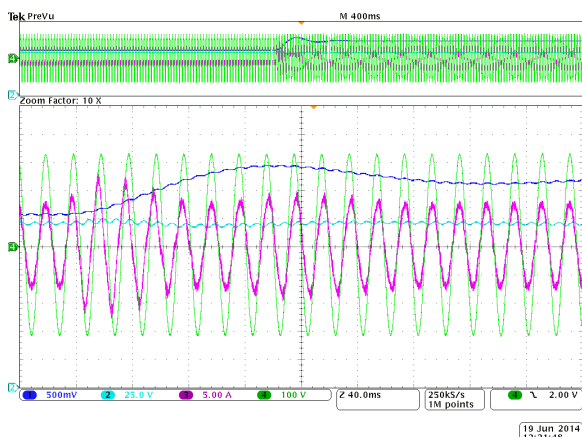
vače proti změně frekvence byla experimentálně ověřena. Výsledky shrnují oscilogramy na obrázcích Obr. 5.25a a 5.25b.

Stejně jako u předchozích topologií měničů i 7L-CHB měnič vyžaduje balancování. Bylo experimentálně ověřeno, že navržená regulační struktura je schopna toto zajistit ve všech provozních stavech měniče. Obr. 5.26a ukazuje odezvu regulačního algoritmu při startu měniče. Následující Obr. 5.26b zobrazuje reakci na skokovou změnu zatížení z 0 na 1 kW. Balancovací algoritmus je schopen udržet napětí na jednotlivých můstcích v rámci přijatelné odchylky (4% v případě můstku H3). Tuto odchylku způsobuje rozdílná kapacita kondenzátoru v meziobvodu H3 (7,2 mF versus 6,8 mF u H1 a H2). Odezva regulace na skokovou změnu zátěže z 1 kW na 0 je na Obr. 5.27b

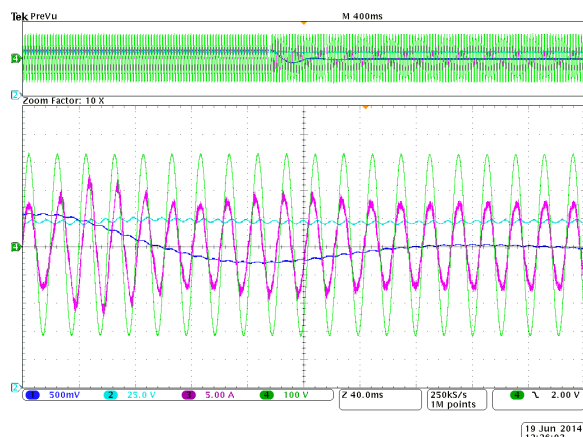
Dále byl ověřen extrémní stav, kdy byl můstek H1 zatížen méně než zbylé dva. Zatížení můstku H1 bylo nastaveno na 90 % můstků H2 a H3. Výsledky experimentu viz Obr. 5.27a pro skok zátěže z 0 na 1 kW. Na průbězích je patrné mírné rozvážení napětí na jednotlivých můstcích, které se však ustálí a dále již neroste. Rozdíl napětí mezi stejně zatíženými můstky H2 a H3 je opět způsoben rozdílnými kapacitami v meziobvodech můstků.

## 5.4. Dílčí závěr

Navržené regulace zkoumaných topologií měničů byly implementovány v číslicovém regulátoru MLC interface a následně experimentálně ověřeny na postavených prototypch. Provedené experimenty potvrdily výborné vlastnosti regulace v ustálených i testovaných přechodných stavech a zároveň byl doložen vhodný návrh MLC interface.

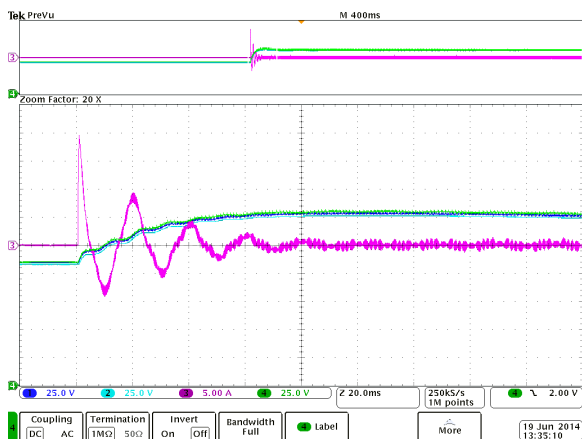


(a) Skok frekvence sítě 50→51 Hz , zátěž 1 kW

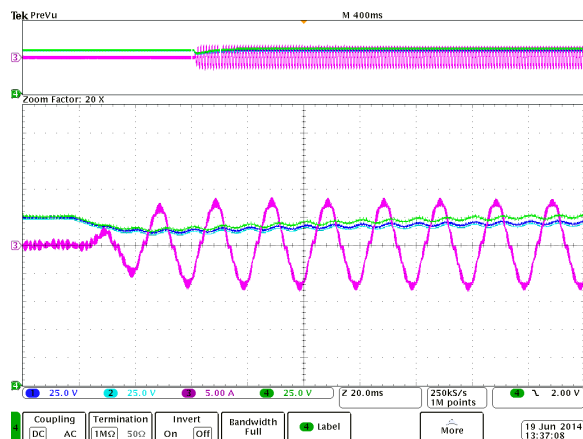


(b) Skok frekvence sítě 50→49 Hz , zátěž 1 kW

Obr. 5.25.: Vliv skoků frekvence fázového napětí sítě na 7L-CHB,  $U_{cw}=450$  V,  $U_{udc1,udc2,udc3}=150$  V, CH1 (modrá):  $\omega$  ze SOGI PLL[0.83 Hz/d], CH2 (světle modrá): napětí meziobvodu můstku H1,  $U_{udc1}$  [25 V/d], CH3 (fialová): fázový proud měniče  $i_{ur}$  [2 A/d], CH4 (zelená): fázové napětí sítě  $u_u$  [100 V/d]

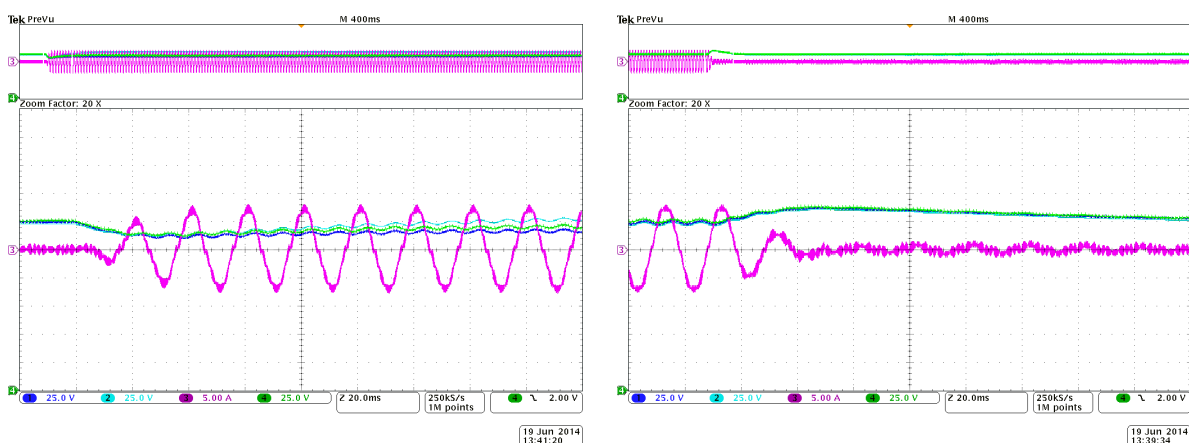


(a) Start nezatíženého usměrňovače



(b) Skok zatížení usměrňovače z 0 na 1 kW

Obr. 5.26.: Experimentální ověření balancování napětí usměrňovače 7L-CHB,  $U_{cw}=450$  V,  $U_{udc1,udc2,udc3}=150$  V,  $f=50$  Hz, CH1 (modrá): napětí meziobvodu můstku H2,  $U_{udc2}$  [25 V/d], CH2 (světle modrá): napětí meziobvodu můstku H1,  $U_{udc1}$  [25 V/d], CH3 (fialová): fázový proud měniče  $i_{ur}$  [2 A/d], CH4 (zelená): napětí meziobvodu můstku H3,  $U_{udc3}$  [25 V/d]



(a) Skok zatížení usměrňovače z 0 na 1 kW

(b) Skok zatížení usměrňovače z 1 kW na 0

Obr. 5.27.: Experimentální ověření balancování napětí na 7L-CHB, nesymetrická zátěž  $P_{H1} = 0,9 \cdot P_{H2}$ ,  $P_{H2} = P_{H3}$ ,  $U_{cw} = 450$  V,  $U_{udc1,udc2,udc3} = 150$  V,  $f = 50$  Hz, CH1 (modrá):  $U_{udc2}$  můstku H2 [25 V/d], CH2 (světle modrá):  $U_{udc1}$  můstku H1 [25 V/d], CH3 (fialová): fázový proud měniče [2 A/d], CH4 (zelená):  $U_{udc3}$  můstku H3 [25 V/d]

Experimentálně bylo ověřeno, že kooperace a synchronizace mezi nadřazeným mikrokontrolérem, který zajišťuje výpočet řídicích algoritmů, a modulátory v FPGA je bezproblémová. Praktickým použitím byla ověřena odolnost číslicového regulátoru vůči vnějším rušivým vlivům vznikajícím při provozu měničů.

Měnič s topologií 3L-NPC (Obr. B.1) je navržen obdobně jako při použití interních PWM generátorů mikrokontroléru, to výrazně zjednodušilo modulátor v FPGA, za cenu větší výpočetní náročnosti regulačního algoritmu, který musí řešit i balancování napětí na kondenzátorech. Experimentálně byla ověřena správnost navržených regulačních algoritmů a PWM generátoru v FPGA. Dalším krokem ve vývoji regulace tohoto měniče je přesun SV PWM modulátoru z mikrokontroléru do hradlového pole. Tato úloha ukazuje, že je pomocí MLC interface možné řídit i externí zařízení, zde řídicí jednotku tyristorového usměrňovače. S výhodou by bylo možno využít pro řízení tyristorového usměrňovače nevyužité PWM výstupy mikrokontroléru nebo přesunout toto řízení do FPGA, které disponuje také volnými PWM výstupy.

Na řízení 4L-FLC měniče (Obr. B.2) bylo ověřeno, že je možné některé z úloh vykonávaných mikrokontrolérem přesunout do hradlového pole, a ušetřit tak výpočetní čas. V tomto případě FPGA zajišťuje aktivní vyvažování napětí na plovoucích kondenzátorech měniče. Pro topologii měniče s plovoucími kondenzátory byla navržena a experimentálně ověřena univerzální startovací procedura, která umožňuje plynulý a bezpečný start tohoto typu měniče přímo z napájecí sítě. Metoda je nezávislá na počtu hladin, ale ukázána je konkrétně pro 4L-FLC.

Na příkladu měniče 7L-CHB (Obr. B.3) byla ukázána realizace kompletního řízení jednofázového napěťového pulzního usměrňovače vhodného pro trakční aplikace včetně jed-

nofázové synchronizace na bázi SOGI PLL. Synchronizace byla vylepšena o řízený start algoritmu, který zajišťuje spolehlivé a rychlé zavěšení fázového závěsu na vstupní napětí.

Pro všechny zkoumané topologie byly provedeny experimenty pro ověření správnosti návrhu regulačních algoritmů. Jedním ze sledovaných parametrů bylo balancování napětí na kondenzátorech, které udržují správné rozložení napětí meziobvodu na výkonové prvky.

## 6. Výpočetně náročné algoritmy v hradlových polích

MLC interface byl navržen také s ohledem na možnost využít FPGA jako pomocný procesor k hlavnímu mikrokontroléru. Už sama topologie vývojové platformy tuto možnost nabízí tím, že lze mapovat registry entit v FPGA do paměťového prostoru mikrokontroléru. Entity akcelerující výpočet pak jsou z pohledu mikrokontroléru transparentní a chovají se jako jakákoli jiná periférie ovšem s pomalejším přístupem k ní vlivem latence externí sběrnice. Také je třeba zohlednit arbitráž této sběrnice, aby nedocházelo ke kolizím a dodatečným prodlevám, neboť tuto sběrnici FPGA sdílí ještě s AD převodníky a systémovým CPLD.

Pro implementaci masivně paralelních algoritmů do FPGA, lze obecně použít dva přístupy:

1. Navrhnout jednu entitu obecně řešící předložený problém a tu pak zduplikovat podle počtu požadovaných výpočtů.
2. Rozdělit výpočet na několik dílčích kroků, ty seřadit za sebe a využít proudového zpracování (pipelining).

První zmiňovaná metoda je vhodná pouze pro jednoduché algoritmy a pouze pro jejich malý počet, typicky se spíše využije u modulárních modulátorů nebo generátorů PWM signálů, které mají jednoduchou topologii. Má vysoké nároky na zdroje FPGA, na druhou stranu máme výsledky dostupné za jeden takt hodin. Topologie MLC interface však neumožňuje efektivně využít současně dostupné výsledky, vždy je nutné přenést je nazpět do mikrokontroléru a tam dále zpracovat.

Pomocí metody využívající proudového zpracování lze výrazně redukovat využití zdrojů FPGA, neboť počítaný algoritmus nejprve rozdělíme na dílčí kroky a zatímco se počítá jeden pro jednu sadu hodnot, mohou předchozí i následující stupně počítat s jinými sadami dat. Výsledky jsou dostupné postupně, což výborně koresponduje s topologií MLC interface, kdy lze přenést výsledek do mikrokontroléru zatímco se počítají další. Navíc proudové zpracování umožní vyšší takt hodinového signálu než kdyby byl výpočet realizován v jednom taktu.

U výpočtů dedikovaných mimo mikrokontrolér je nutno zohlednit dopad rychlosti přenosu, resp. cyklu externí sběrnice, který je minimálně 100 ns, na celkovou dobu výpočtu. Za jeden cyklus sběrnice je možné přenést 16 bitů (tedy 2 B dat), což v případě velkého množství přenášených dat vede na výraznou dodatečnou prodlevu, se kterou je třeba počítat a zvážit, zda se vůbec vyplatí výpočet provádět pomocí FPGA.

Vývoj, resp. návrh složitých algoritmů pro výpočet v hradlovém poli je komplikovaný, využívají se tedy nástroje, které umožňují vývoj urychlit. Z komerčně dostupných se používá např. kombinace Matlab Simulinku spolu s vývojovým prostředím Quartus, pro které existuje rozšíření Altera DSP builder [75]. To umožňuje převést model z Matlabu/Simulinku do formy projektu pro Quartus (tedy VHDL nebo Verilog) a následně navržený model zkompileovat pro hradlového pole. Obecnou funkcí Matlabu je Simulink HDL coder, který umožňuje bloky Simulinku převést do VHDL nebo Verilogu.

Z volných nástrojů (open source) je dostupný např. program FloPoCo (Floating-Point Cores) [76], který umožňuje generovat dílčí bloky pro vybrané matematické operace (+, -, \*, / atd.). Na uživateli je vygenerované dílčí bloky spojit do finálního návrhu, vložit do projektu vývojového prostředí pro použitý obvod, zkompileovat a nahrát. Existuje samozřejmě mnoho dalších nástrojů urychlujících vývoj kódu pro hradlová pole.

Jedním ze zástupců výpočetně extrémně náročných algoritmů, na jejichž vývoji autor pracoval a které byly dotaženy pro úlohy bezsenzorového řízení střídavých pohonů až do laboratorních prototypů, je marginalizovaný částicový filtr (MPF). Tento stochastický filtr kombinuje Kalmanovy filtry s částicovými filtry, tzn. každý particle (částice) je tvořen Kalmanovým filtrem. Rozkládá tak řešený problém na dvě části: lineární a nelineární. Kalmanův filtr pak řeší lineární část a částicový filtr nelineární část modelu. To umožňuje přesně reprezentovat nelineární model motoru v celém rozsahu otáček včetně nulových.

Výpočetní náročnost tohoto algoritmu lineárně roste s počtem počítaných částic, samozřejmě větší počet částic znamená obecně i větší přesnost estimace. Obdobná situace je u prediktivních regulačních algoritmů: mnoho kroků výpočtu jen s jinými parametry. Implementace tohoto druhu výpočtů pro mikrokontrolér není často nijak problematická, avšak je neefektivní a náročná na jeho výpočetní výkon. Naproti tomu jejich implementace do FPGA je velmi efektivní, jednodušší algoritmy lze počítat paralelně, u složitějších se využívá proudové zpracování (pipelining), což sice prodlužuje dobu výpočtu, na druhou stranu jsou výrazně ušetřeny zdroje dostupné v FPGA.

## 6.1. Návrh marginalizovaného částicového filtru pro PMSM

Pro ověření možnosti akcelarovat výpočet pomocí FPGA byl vybrán marginalizovaný částicový filtr, zde použitý pro bezsenzorové zjišťování polohy a rychlosti rotoru synchronního stroje s permanentními magnety. Tento výpočet je velmi náročný, vliv doby přenosu dat do/z FPGA se na celkové době výpočtu příliš negativně neprojeví. Výsledky tohoto výzkumu byly prezentovány v [35] a [77], dále budou shrnuty části podstatné pro implementaci do hradlového pole.

### 6.1.1. Matematický model PMSM

Běžně používaný model motoru je matematický model v rotujícím souřadném systému, který je svázaný s rotorovým tokem. Je diskretizován pomocí Eulerovy metody prvního řádu pro časový krok  $\Delta t$ :

$$i_{d,t+1} = a_d i_{d,t} + b_d i_{q,t} \omega_{me,t} + c_d u_{d,t} + \epsilon_{d,t} \quad (6.1)$$

$$i_{q,t+1} = a_q i_{q,t} - f_q \omega_{me,t} - b_q i_{d,t} \omega_{me,t} + c_q u_{q,t} + \epsilon_{q,t} \quad (6.2)$$

$$\omega_{me,t+1} = \omega_{me,t} + \epsilon_{\omega,t} \quad (6.3)$$

$$\vartheta_{e,t+1} = \vartheta_{e,t} + \omega_{me,t} \Delta t + \epsilon_{\vartheta,t} \quad (6.4)$$

kde

$i_{d,t}, i_{q,t}$	... složka statorového proudu v ose $d$ , resp. $q$
$u_{d,t}, u_{q,t}$	... složka statorového napětí v ose $d$ , resp. $q$
$a_d, a_q, b_d, b_q, c_d, c_q, f_q$	... konstanty modelu pro přehlednější zápis
$\omega_{me,t}$	... elektrická úhlová rychlost
$\vartheta_{e,t}$	... elektrická poloha rotoru
$\epsilon_{d,t}, \epsilon_{q,t}, \epsilon_{\omega,t}, \epsilon_{\vartheta,t}$	... šumové složky příslušných veličin
$\Delta t$	... perioda vzorkování

Konstanty použité pro zpřehlednění zápisu rovnic modelu výše jsou rozepsány níže, rovnice (6.5) až (6.11).

$$a_d = \left( 1 - \frac{R_s}{L_{sd}} \Delta t \right) \quad (6.5)$$

$$a_q = \left( 1 - \frac{R_s}{L_{sq}} \Delta t \right) \quad (6.6)$$

$$b_d = \frac{L_{sq}}{L_{sd}} \Delta t \quad (6.7)$$

$$b_q = \frac{L_{sd}}{L_{sq}} \Delta t \quad (6.8)$$

$$c_d = \frac{\Delta t}{L_{sd}} \quad (6.9)$$

$$c_q = \frac{\Delta t}{L_{sq}} \quad (6.10)$$

$$f_q = \frac{\Psi_{pm}}{L_{sq}} \Delta t \quad (6.11)$$

kde

$R_s$	...	statorový odpor
$L_{sd}, L_{sq}$	...	statorová indukčnost v ose $d$ , resp. $q$
$\Delta t$	...	vzorkovací perioda
$\psi_{pm}$	...	tok vybuzený permanentními magnety na rotoru

Rovnice (6.3) je zjednodušená s použitím předpokladu, že je počítáno s velkým momentem setrvačnosti motoru v porovnání se vzorkovací periodou (tato zjednodušující podmínka je většinou splněna). Šumové složky jsou zastoupeny členy  $\epsilon_{d,t}$ ,  $\epsilon_{q,t}$ ,  $\epsilon_{\omega,t}$  a  $\epsilon_{\vartheta,t}$ . Ty v sobě zahrnují chyby způsobené nepřesnou diskretizací, změnami parametrů (např. vlivem teploty, saturace magnetického obvodu) a také nezahrnutými vlivy (jako neznámá zátěž, vliv mrtvých časů, nelineární úbytky na polovodičích atp.).

Byl použit redukovaný model PMSM, tzn. stavový vektor má tvar  $x_t = [\omega_{me,t}, \vartheta_{e,t}]$ . Stavové rovnice jsou pak (6.3), (6.4) a pro pozorování slouží dvojice (6.1) a (6.2). V bezsenzorovém módu jsou pozorovány dvě veličiny, a to proudy v souřadném systému  $\alpha$ - $\beta$ , které dále přepočteme do rotujícího souřadného systému  $d$ - $q$ , pomocí rovnic (6.12) a (6.13). Chyby měření jsou zahrnuty v členech  $\bar{\epsilon}_{d,t}$  a  $\bar{\epsilon}_{q,t}$ .

$$\bar{i}_{d,t} = \bar{i}_{\alpha,t} \cos \vartheta_{e,t} + \bar{i}_{\beta,t} \sin \vartheta_{e,t} + \bar{\epsilon}_{d,t} \quad (6.12)$$

$$\bar{i}_{q,t} = -\bar{i}_{\alpha,t} \sin \vartheta_{e,t} + \bar{i}_{\beta,t} \cos \vartheta_{e,t} + \bar{\epsilon}_{q,t} \quad (6.13)$$

kde

$\bar{i}_{d,t}, \bar{i}_{q,t}$	...	predikované proudy v osách $d$ , resp. $q$
$\bar{i}_{\alpha,t}, \bar{i}_{\beta,t}$	...	predikované proudy v osách $d$ , resp. $q$
$\bar{\epsilon}_{d,t}, \bar{\epsilon}_{q,t}$	...	šumové složky příslušných veličin

### 6.1.2. Marginalizovaný částicový filtr

Algoritmus MPF je možné použít pro problémy, které je možné rozložit na lineární ( $x_t^l$ ) a nelineární část ( $x_t^n$ ) a tedy  $x_t = [x_t^l, x_t^n]$ . Například:

$$x_{t+1}^l = A(x_t^n)x_t^l + \epsilon_{l,t} \quad (6.14)$$

$$x_{t+1}^n = f(x_t, \epsilon_{n,t}) \quad (6.15)$$

$$y_t = C(x_t^n)x_t^l + \epsilon_{y,t} \quad (6.16)$$



Pokud je známa hodnota  $x_t^n$ , pak rovnice (6.14) a (6.16) reprezentují lineární Gaussovský model, který může být odhadován pomocí Kalmanova filtru. Výsledný odhad bude Gaussovské rozložení se střední hodnotou a kovariancí závislých na nelineární části (ta je známa). Nelineární část reprezentují částice (partikly)  $x_t^{n(i)}$ , kde  $i = 1, \dots, n$  jsou vzorky z nelineární části. Lineární části jsou odhadovány Kalmanovými filtry, se střední hodnotou  $\hat{x}_t^{l(i)}$  a kovarianční maticí  $P_t^{(i)}$  pro každou z částic.

### 6.1.3. Marginalizovaný částicový filtr pro PMSM

Pro známou hodnotu  $\vartheta_{e,t}$  je v modelu PMSM stavová proměnná reprezentována hodnotou  $\omega_{me,t}$ , kdy rovnice (6.1) a (6.2) reprezentují pozorovatele a (6.3) reprezentuje stavovou část modelu. Model (6.1) - (6.4) lze přepsat do formy (6.14)-(6.16) přiřazením:

$$A = 1 \quad (6.17)$$

$$y_t = [i_{d,t} - a_d i_{d,t-1} - c_d u_{d,t-1}, i_{q,t} - a_q i_{q,t-1} - c_q u_{q,t-1}]^T \quad (6.18)$$

$$C = [b_d i_{q,t}, -(f_q + b_q i_{d,t})]^T \quad (6.19)$$

Aplikováním rovnic standardního Kalmanova filtru na rovnice výše jsou získány rovnice pro odhad rychlosti rotoru pro danou částici pozice rotoru  $\vartheta_{e,t}^{(i)}$  níže:

$$\hat{\omega}_{me,t}^{(i)} = \hat{\omega}_{me,t-1}^{(i)} + K^{(i)} (y_t - C^{(i)} \hat{\omega}_{me,t-1}^{(i)}) \quad (6.20)$$

$$K^{(i)} = P_{t-1}^{(i)} C^{(i)T} \rho^{(i)}$$

$$\rho^{(i)} = \frac{1}{r} (1 - \zeta^{(i)} C^{(i)T} C^{(i)})$$

$$\zeta = \frac{P_{t-1}^{(i)}}{r + P_{t-1}^{(i)} C^{(i)T} C^{(i)}}$$

$$P_t^{(i)} = P_{t-1}^{(i)} (1 - K^{(i)} C^{(i)}) + q_\omega \quad (6.21)$$

Každý takto vypočtený filtr má dále svou váhu:

$$\hat{w}_t^{(i)} \propto \sqrt{\frac{\rho}{r}} \exp\left(-\frac{1}{2}(y - \hat{y})' \left(\frac{1}{r}(I - \zeta C C')\right)(y - \hat{y})\right) w_{t-1}^{(i)} \quad (6.22)$$

Protože rekurzivní výpočet  $\hat{w}_t^{(i)}$  (6.22) konverguje u jedné částice k jedné a u ostatních k nule, je potřeba provádět převzorkování. Převzorkování však přidává do výsledků další stochastické variance, je vhodné jej provádět pouze v nezbytných případech. Pro rozhodování zda provést nebo neprovést převzorkování výsledků, je použita podmínka na efektivní velikost vzorku, např.  $n_{eff} < 0,9n$ . Efektivní velikost vzorku je spočtena dle rovnice (6.23).

$$n_{eff} = \frac{1}{\sum_{i=1}^n \left(w_t^{(i)}\right)^2} \quad (6.23)$$

Postup výpočtu algoritmu pro PMSM je následující:

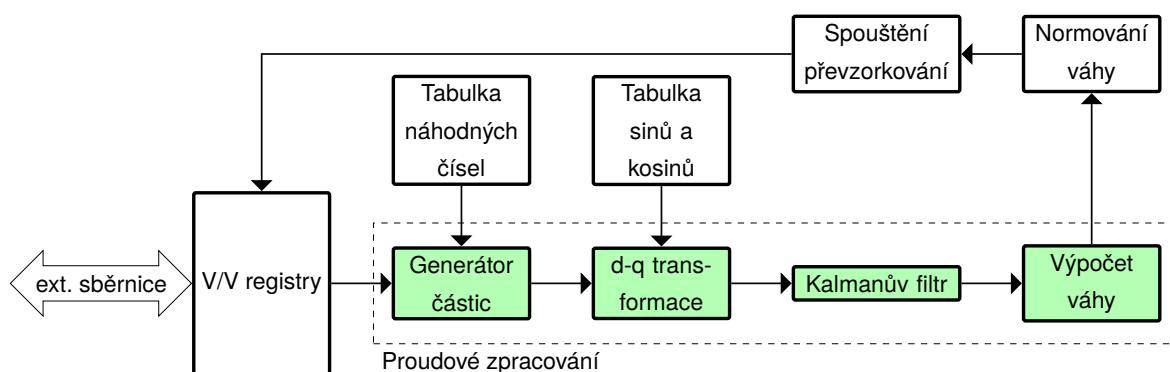
- Inicializace: vygenerují se  $\omega_0^{(i)}$ ,  $\vartheta_0^{(i)}$  a nastaví se počáteční váhy částic na hodnotu  $w_0^{(i)} = 1/N$ .
- Při běhu algoritmu se v každém vzorkovacím kroku spočte:
  1. Pro každou z částic:
    - a) Vygeneruje se nová hodnota  $\vartheta_t^{(i)}$  pomocí rovnice (6.4).
    - b) Vypočte se transformace proudů  $i_{a,t}$  a  $i_{b,t}$  do souřadného systému  $d-q$  podle (6.12) a (6.13).
    - c) Spočte se Kalmanův filtr, rovnice (6.20) a (6.21), tím se získají hodnoty  $\hat{\omega}_t^{(i)}$  a  $P_t^{(i)}$ .
    - d) Jako poslední krok se vypočte nenormalizovaná váha  $\hat{w}_t$ , podle (6.22).
  2. Znormují se jednotlivé váhy částic,  $w_t = \hat{w}_t / \sum_{i=1}^n \hat{w}_t$ .
  3. Částice se deterministicky převzorkují.

Výsledkem výše uvedených výpočtů je sada výsledků s různými váhami a pro regulaci v uzavřené smyčce je třeba jedna konkrétní hodnota, je nutno tuto hodnotu vybrat. Jedna z možností je využít již spočtené váhy částic a jako výsledek vzít tu s nejvyšší vahou,  $w_t^{i_{best}} \geq w_t^i, \forall i$ . Odhadnutá rychlost rotoru je pak  $\hat{\omega}_{me,t} = \hat{\omega}_{me,t}^{(i_{best})}$  a poloha je  $\hat{\vartheta}_{me,t}^{(i_{best})}$ .

#### 6.1.4. Implementace MPF do FPGA

Původní implementace byla realizována pouze v mikrokontroléru typu TMS320F28335 a výpočet pěti těchto filtrů trval  $47 \mu s$  při hodinové frekvenci mikrokontroléru 150 MHz. Podrobnosti jsou uvedeny v práci [35], která prezentuje první negaussovský filtr aplikovaný v oblasti bez-senzorového řízení střídavých elektrických pohonů. Výše zmíněná doba výpočtu je akceptovatelná pro regulační smyčky s nižší vzorkovací frekvencí, avšak např. pro hysterezní řízení, DTC nebo FCS-MPC je doba výpočtu kritický parametr, který negativně ovlivňuje kvalitu regulace. Přenesením výpočtu tohoto algoritmu do FPGA lze výpočetní čas výrazně zkrátit.

Implementace částicového filtru je navržena jako proudově zpracovávaná (pipeline), s výjimkou normalizace váhy částic a převzorkování, má 61 kroků a všech 83 výpočtů se provádí jednou za takt hodin entity. Blokové schéma viz Obr. 6.1. Výpočet trvá minimálně 72 tiků hodin (pro jednu částici) a každá další počítaná částice prodlužuje dobu výpočtu o dva takty. Hodinový signál je nastaven na 35 MHz, čemuž odpovídá výpočetní čas  $2,3 \mu s$ , pokud



Obr. 6.1.: Blokové schéma algoritmu pro výpočet MPF v FPGA

je zahrnuta i doba nutná na přenos dat do/z FPGA, vychází hodnota  $3 \mu s$ , v obou případech je počítáno šest částic.

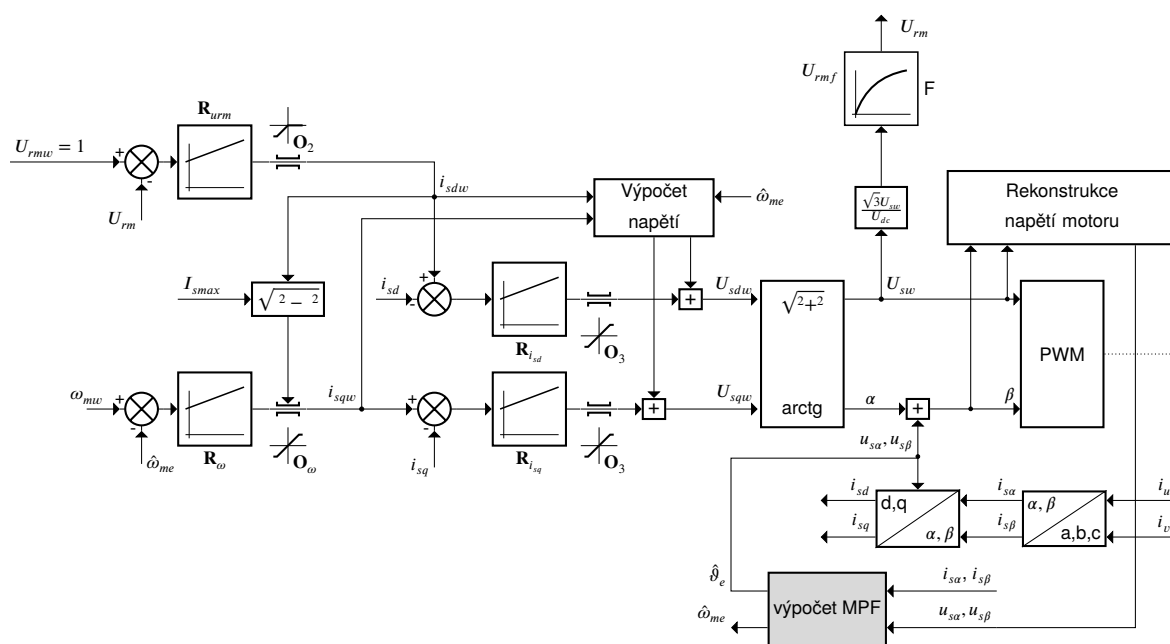
Výpočty se provádí v plovoucí řádové čárce s omezeným rozsahem, 5 bitů je použito pro exponent a 16 pro mantisu. Tato optimalizace ušetří zdroje hradlového pole i za cenu nižšího dynamického rozsahu a přesnosti.

Blokové schéma navrženého filtru (Obr. 6.1) zobrazuje jednotlivé části implementace. Pomocí bloku *V/V registry* je zajištěna komunikace s okolím, resp. mikrokontrolérem, umožňuje nastavovat vstupní proměnné, číst výsledky a spouštět výpočet. Následuje generátor částic, který zajišťuje "označování" jednotlivých částic procházejících proudovým zpracováním, aby bylo možno detekovat konec výpočtu. Ve výpočtu je použito o dvě částice více, jedna prochází výpočtem jako první a protože do ní zapisují všechny kroky výpočtu, neobsahuje relevantní data. Druhá přidaná částice je zpracovávána jako poslední a slouží především pro detekci dokončení výpočtu. Druhou funkcí generátoru je přidání pseudo náhodných hodnot  $q_{\theta} \epsilon_{\theta}$  ke každé částici  $\vartheta_e^{(i)}$  z tabulky náhodných čísel. Tato tabulka je implementována jak ROM paměť, obsahující hodnoty vybrané pravděpodobnostní distribuční funkce (v tomto případě se jedná o 997 hodnot). Každý průchod částice generátorem způsobí posun ukazatele v této tabulce na následující hodnotu.

Výpočet funkcí sinus a kosinus je implementován pomocí tabulky, kde je uloženo 8192 předpočítaných hodnot reprezentujících jednu čtvrtinu funkce sinus.

Blok Kalmanova filtru vypočítává rovnice (6.20) a (6.21). Výpočet váhy je nejnáročnější částí algoritmu především kvůli funkci  $\exp$  (rovnice (6.22)), která byla navržena za pomoci nástroje MegaWizard. Tento blok zároveň vybírá nejpravděpodobnější výsledek s indexem  $i_{best}$ . Normalizace vah jednotlivých částic je již mimo proudové zpracování, neboť k jejímu výpočtu je třeba znát ji u všech částic.

Posledním blokem je spouštění převzorkování, ten signalizuje nadřazenému mikrokontroléru nutnost provést převzorkování výsledných vah u jednotlivých částic. Mikrokontrolér pak rozhodne, která z částic bude vyřazena a která se zduplikuje a bude pokračovat ve výpočetním cyklu.



Obr. 6.2.: Blokové schéma regulačního algoritmu s blokem výpočtu MPF v FPGA

### 6.1.5. Experimentální výsledky

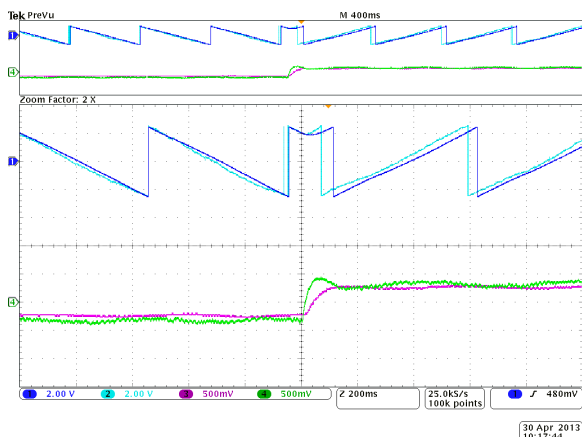
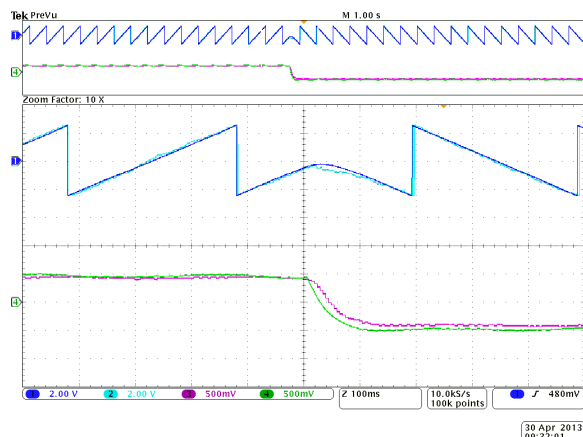
Blokové schéma zkoumaného bezsenzorového pohonu je zobrazeno na Obr. 6.2. Algoritmus řízení pohonu je založen na vektorovém řízení v Kartézských souřadnicích v rotujícím souřadném systému  $d-q$ .

Do bloku Výpočet MPF vstupují statorové proudy  $i_{s\alpha}$  a  $i_{s\beta}$ , spolu s rekonstruovaným vektorem statorového napětí  $u_{s\alpha}$  a  $u_{s\beta}$ . Ten je rekonstruován z napětí meziobvodu  $U_{dc}$  a známé kombinace sepnutí napěťového střídače. Výstupem MPF bloku je pak odhadovaná pozice rotoru  $\hat{\vartheta}_e$  a jeho rychlost  $\hat{\omega}_{me}$ .

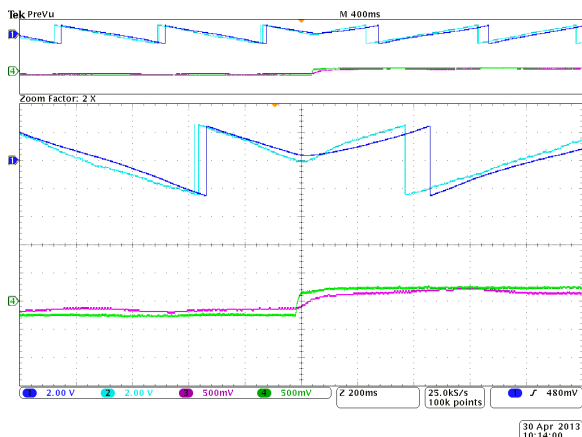
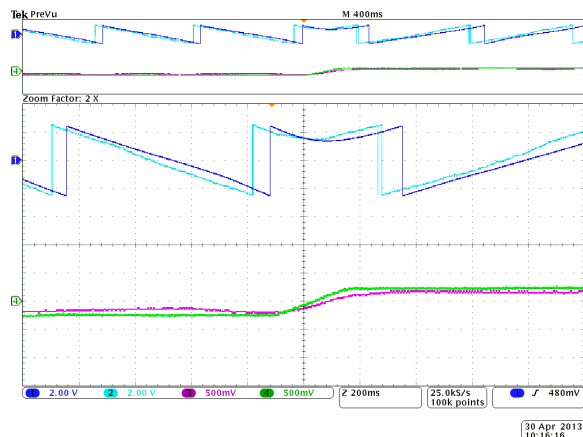
Experimenty probíhaly na soustrojí s PMSM o jmenovitém výkonu 10,7 kW. Byly provedeny testy řízení jak se senzorem, tak bez něho. Všechny výsledky jsou měřeny pro algoritmus MPF s šesti částicemi a kovarianční matice byly nastaveny na:  $q_w = 0, 1$ ,  $q_\theta = 0, 003$ ,  $r = 0, 05$ .

První experiment byl proveden v režimu s čidlem otáček, s obdélníkovým rychlostním profilem měnicím požadavek na otáčky v rozmezí  $\pm 30 \text{ ot. min}^{-1}$  (Obr. 6.3a) nebo  $\pm 50 \text{ ot. min}^{-1}$  (Obr. 6.3b). Odhady rychlosti i polohy rotoru jsou dostatečně spolehlivé s chybou do 10 stupňů (v případě polohy).

Druhý experiment byl proveden v bezsenzorovém režimu, kdy výstup z čidla nahradily odhadované hodnoty s největší pravděpodobností  $\hat{\vartheta}_e^{(i_{best})}$  a  $\hat{\omega}_{me}^{(i_{best})}$ . Porovnání odhadovaných hodnot s naměřenými čidlem ukazuje pro skokovou změnu požadavku Obr. 6.4a a pro změnu po rampě Obr. 6.4b. Tyto experimenty ukazují, že je pohon schopen spolehlivé funkce od  $20 \text{ ot. min}^{-1}$  při chybě odhadu polohy do 25 stupňů.

(a) Skok z  $-30 \text{ ot. min}^{-1}$  na  $+30 \text{ ot. min}^{-1}$ (b) Skok z  $-50 \text{ ot. min}^{-1}$  na  $+50 \text{ ot. min}^{-1}$ 

Obr. 6.3.: Řízení PMSM s čidlem polohy, MPF v otevřené smyčce, skoková změna požadavku na rychlost, CH1 (modrá): elektrická poloha rotoru (ze senzoru) [144 st./d], CH2 (světle modrá): odhadovaná elektrická poloha rotoru (z MPF) [144 st./d], CH3 (fialová): rychlost rotoru (ze senzoru) [50 ot.min<sup>-1</sup>/d], CH4 (zelená): odhadovaná rychlost rotoru (z MPF) [50 ot.min<sup>-1</sup>/d]

(a) Skok z  $-30 \text{ ot. min}^{-1}$  na  $+30 \text{ ot. min}^{-1}$ (b) Přechod z  $-30 \text{ ot. min}^{-1}$  na  $+30 \text{ ot. min}^{-1}$  po rampě

Obr. 6.4.: Řízení PMSM bez čidla polohy, změna požadavku na rychlost, CH1 (modrá): elektrická poloha rotoru (ze senzoru) [144 st./d], CH2 (světle modrá): odhadovaná elektrická poloha rotoru (z MPF) [144 st./d], CH3 (fialová): rychlost rotoru (ze senzoru) [50 ot.min<sup>-1</sup>/d], CH4 (zelená): odhadovaná rychlost rotoru (z MPF) [50 ot.min<sup>-1</sup>/d]

## 6.2. Dílčí závěr

Na příkladu výpočtu marginalizovaného částicového filtru bylo experimentálně ověřeno, že hradlové pole přítomné na MLC interface lze použít i pro akceleraci náročných výpočtů, které by jinak (při výpočtu v mikrokontroléru) znemožnily použití některých řídicích technik, např. přímého řízení momentu.

V porovnání s implementací v mikrokontroléru je výpočet v hradlovém až 15 x rychlejší (47  $\mu s$  versus 3  $\mu s$ ). Takováto redukce výpočetního času umožňuje využít i toto komplexní a výpočetně náročné bezsensorové řízení v aplikacích s velkými požadavky na rychlost regulační smyčky, jako je např. přímé řízení momentu.

## 7. Podpora rapid prototypingu pro MLC interface - Model Based Design (MBD)

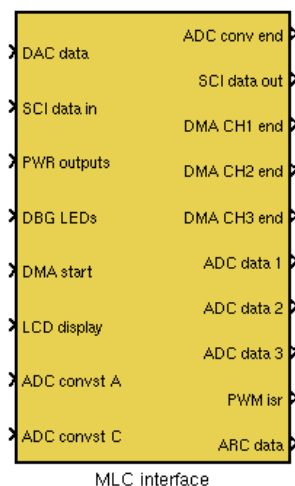
Vývoj složitých výpočetních a regulačních algoritmů včetně řízení výše zmiňovaných víceúrovňových topologií spočívá v prvotním simulačním ověření daného problému a následnými experimenty. Experimenty jsou podmíněny převedením simulačního schématu/modelu do vhodného programovacího jazyka pro cílovou platformu (u mikrokontrolérů jde většinou o jazyk C). Proces převodu simulace do fáze experimentu lze urychlit pomocí nástrojů pro rapid prototyping. Ty automaticky generují zdrojový kód z navrženého simulačního schématu. Např. program Matlab podporuje generování kódu v jazyce C podle navrženého regulačního schématu pro vybrané mikrokontrolérové platformy i pro hradlová pole. Vhodnou volbou mikroprocesoru i hradlového pole bylo dosaženo bezproblémové podpory této funkce i pro vyvinutý mikroprocesorový regulátor MLC interface.

### 7.1. Blok MLC interface pro Simulink

Nadstavba programu Matlab Simulink přímo podporuje mikrokontroléry s jádrem řady C2000, mezi které patří i obvody TMS320F2812 a TMS320F28335, použitelné ve formě mikroprocesorových modulů spolu s MLC interface. Bylo třeba navrhnout doplňkové bloky pro Simulink, které by umožnily využít externí periferie na MLC interface, především analogově-digitální převodníky.

Topologie MLC interface ulehčila vývoj bloku, kdy je k většině periferií MLC interface přístupováno pomocí zápisu na adresu příslušné periferie (pomocí Memory Copy bloku Simulinku). Navržený blok reprezentující MLC interface (Obr. 7.1) poskytuje veškerou funkčnost nutnou pro návrh regulační struktury a následné vygenerování zdrojového kódu v jazyce C. Blok podporuje pouze mikrokontrolér TMS320F28335, protože TMS320F2812 nepodporuje DMA přenosy a TMS570LS3137 není podporován Simulinkem. Mezi klíčové vlastnosti bloku patří:

- Ruční spouštění převodu analogově digitálních převodníků (*ADC convst A, C*) nebo automatické spouštění analogově digitálních převodníků pomocí ePWM bloku.
- Přednastavené DMA přenosy dat z převodníků do paměti .



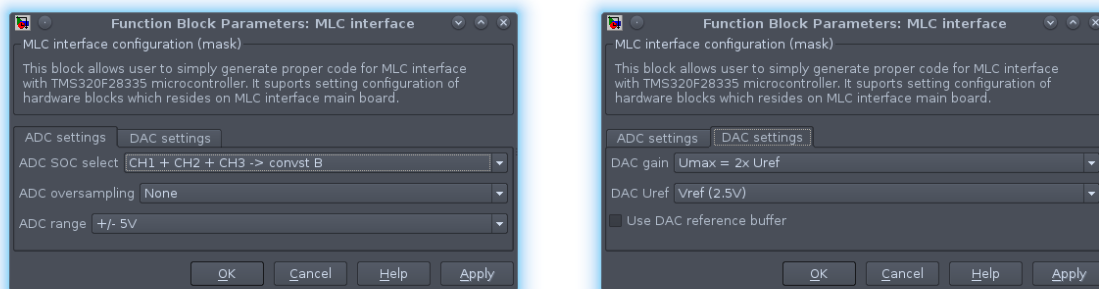
Obr. 7.1.: Navržený blok pro ovládání MLC interface s TMS320F28335 v Matlab/Simulinku

- Ruční spuštění přenosu dat z analogově digitálních převodníků (*DAM start*) bez naměření nových hodnot.
- Výstup analogových hodnot pomocí digitálně analogového převodníku (*DAC data*).
- Výstup na LED připojené k LCD (*DBG LEDs*).
- Výstup na znakový LCD displej (*LCD display*).
- Komunikace přes sériovou linku (*SCI data in* a *SCI data out*).
- Ovládání relé apod. přes PWR výstupy (*PWR outputs*).
- Konfigurace analogově digitálních a digitálně analogových převodníků je realizována v grafickém rozhraní.

Popis vstupů a výstupů navrženého bloku pro Simulink shrnují tabulky Tab. 7.1 a 7.2. Nastavení bloku je realizováno přes grafické rozhraní (Obr. 7.2), pomocí kterého je možné nastavit analogově-digitální i digitálně-analogové převodníky.

U A/D převodníků lze zvolit mapování spouštěcích vstupů (*ADC convst A, B, C*) na jednotlivé převodníky. Pro ruční spouštění převodu se používají vstupy *ADC convst A* a *ADC convst C*, které se liší aktivační hranou. Spouštěcí vstup *ADC convst B* využívá blok vnitřně pro hardwarové spouštění převodu za pomoci ePWM generátoru dostupného na mikrokontroléru. Oba druhy spouštění lze libovolně kombinovat, avšak po dokončení převodu se přenášejí vždy data ze všech tří analogově digitálních převodníků i v případě, že nová data má jen jeden nebo dva z nich. Stejně tak pulz na vstupu *DMA start* spustí DMA přenos (bez měření) pro všechny tři převodníky. Kromě spouštění převodu je možné nastavit převzorkování vstupního signálu, které však násobně prodlouží dobu převodu převodníků. Poslední





Obr. 7.2.: Konfigurační dialog bloku MLC interface v Simulinku

Tab. 7.1.: Význam vstupních portů bloku MLC interface

Jméno vstupu	Parametry	Funkce
DAC data	pole osmi <i>uint16_t</i> hodnot	S každou iterací vzorkovací periody vypíše na výstup jednu hodnotu z pole vstupních hodnot. Hodnota s indexem 1 se zapíše na výstup A digitálně analogového převodníku, 2 na B atd.
SCI data in	jeden byte	V každém průchodu vzorkovací smyčky odešle po sériové lince hodnotu na svém vstupu.
PWR outputs	<i>uint16_t</i>	Odešle hodnotu svého vstupu na adresu PWR výstupů vyvedených ze systémového CPLD každou vzorkovací periodu. Platných je spodních 8 bitů.
DBG LEDs	<i>uint16_t</i>	Hodnotu spodních pěti bitů ze vstupní hodnoty zapíše na DEBUG výstupy systémového CPLD.
DMA start	<i>uint16_t</i>	Ruční spuštění přenosu dat z analogově-digitálních převodníků. Ten se aktivuje při přechodu toho signálu z 0 kladnou hodnotu.
LCD display	pole 32 znaků reprezentovaných jako <i>uint_16t</i>	Periodicky odesílá na LCD displej MLC interface hodnoty zapsané ve vstupním poli.
ADC convst A	<i>uint16_t</i>	Ruční spuštění analogově-digitálních převodníků. Ty se aktivují při přechodu toho signálu z 0 do kladné hodnoty, tj. pozitivní logika.
ADC convst C	<i>uint16_t</i>	Ruční spuštění analogově-digitálních převodníků. Ty se aktivují při přechodu toho signálu z kladných hodnot na 0, tj. negativní logika.

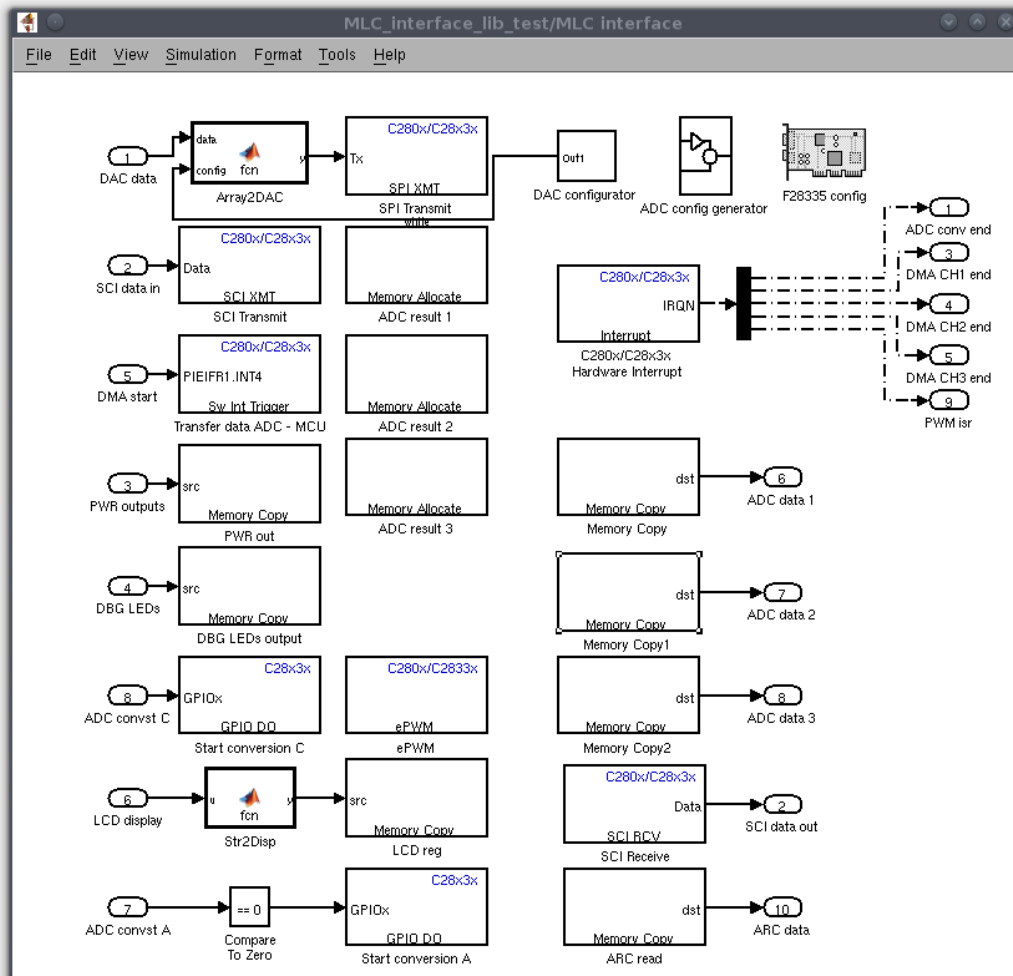
volba na dialogu poskytuje možnost zvolit vstupní rozsah ADC z možných  $\pm 5\text{ V}$  nebo  $\pm 10\text{ V}$ . Všechny konfigurační volby se aplikují na každý z převodníků stejně, oddělená konfigurace není momentálně podporována.

Tab. 7.2.: Význam výstupních portů bloku MLC interface

Jméno výstupu	Parametry	Funkce
ADC conv end	obsluha přerušení	Na tento signál lze napojit obslužnou rutinu pro přerušení od konce převodu analogově-digitálních převodníků. Tento signál je asynchronní k běhu všech vzorkovacích smyček modelu.
SCI data out	jeden byte	V každém průchodu vzorkovací smyčky načte po sériové lince odeslanou hodnotu.
DMA CH1 end	obsluha přerušení	Asynchronní signál generovaný po dokončení přenosu dat z prvního kanálu analogově-digitálních převodníků.
DMA CH2 end	obsluha přerušení	Asynchronní signál generovaný po dokončení přenosu dat z druhého kanálu analogově-digitálních převodníků.
DMA CH3 end	obsluha přerušení	Asynchronní signál generovaný po dokončení přenosu dat z třetího kanálu analogově-digitálních převodníků.
ADC data 1	pole osmi hodnot <i>int16_t</i>	Výstup naměřených dat, jsou odstraněny dva nejméně významné bity, hodnota pole s indexem 1 odpovídá vstupu 0 ADC CH1.
ADC data 2	pole osmi hodnot <i>int16_t</i>	Výstup naměřených dat, jsou odstraněny dva nejméně významné bity, hodnota pole s indexem 1 odpovídá vstupu 0 ADC CH2.
ADC data 3	pole osmi hodnot <i>int16_t</i>	Výstup naměřených dat, jsou odstraněny dva nejméně významné bity, hodnota pole s indexem 1 odpovídá vstupu 0 ADC CH3.
PWM isr	obsluha přerušení	Asynchronní signál generovaný PWM modulem použitým pro hardwarové spouštění převodu analogově-digitálních převodníků.
ARC data	<i>uint16_t</i>	Hodnota načtená z absolutního čidla polohy, kódovaná Grayovým kódem.

V konfigurační záložce D/A převodníků lze nastavit rozsah jejich výstupního napětí jako jedno nebo dvojnásobek referenčního napětí. Zvolit referenční napětí 2,5 V nebo 5 V. Volba referenčního napětí 5 V automaticky nastavuje rozsah výstupního napětí na 1x referenční (to je vlastnost použitého obvodu). Dále lze povolit posilovač referenčního napětí, který snižuje proudové zatížení zdroje referenčního signálu.

Vnitřní struktura bloku je ukázána na Obr. 7.3. Pro komunikaci s periferiemi na hlavní desce MLC interface je využito Simulink bloků *Memory Copy*, které mají zadány příslušné adresy periferií (např. *PWR out*, *DBG LEDs output* atd.). Bloky *Memory Allocate* zajišťují alokaci polí o velikosti 16 slov pro ukládání dat přenesených z A/D převodníků (ADC result 1,2,3). Na ně jsou navázány bloky *Memory Copy*, které zajišťují výstup pole o osmi hodnotách



Obr. 7.3.: Vnitřní blokové schéma bloku pro MLC interface

s naměřenými hodnotami ven z bloku (není použito druhé slovo z A/D převodníků s dvěma platnými bity).

Bloky *DAC configurator* a *ADC config generator* zapisují uživatelem zvolené nastavení převodníků do příslušných konfiguračních registrů periferií. Obě periferie se nastavují pouze na začátku běhu vygenerovaného programu. Vstupní hodnoty pro D/A převodník jsou zpracovány pomocí Matlab funkce *Array2DAC*. Ta zajistí rozložení vstupního pole o osmi hodnotách pro DAC na jednotlivé zápisy do registrů výstupů převodníku přes SPI sběrnici, zároveň při prvním spuštění zapíše vygenerované konfigurační slovo do DAC.

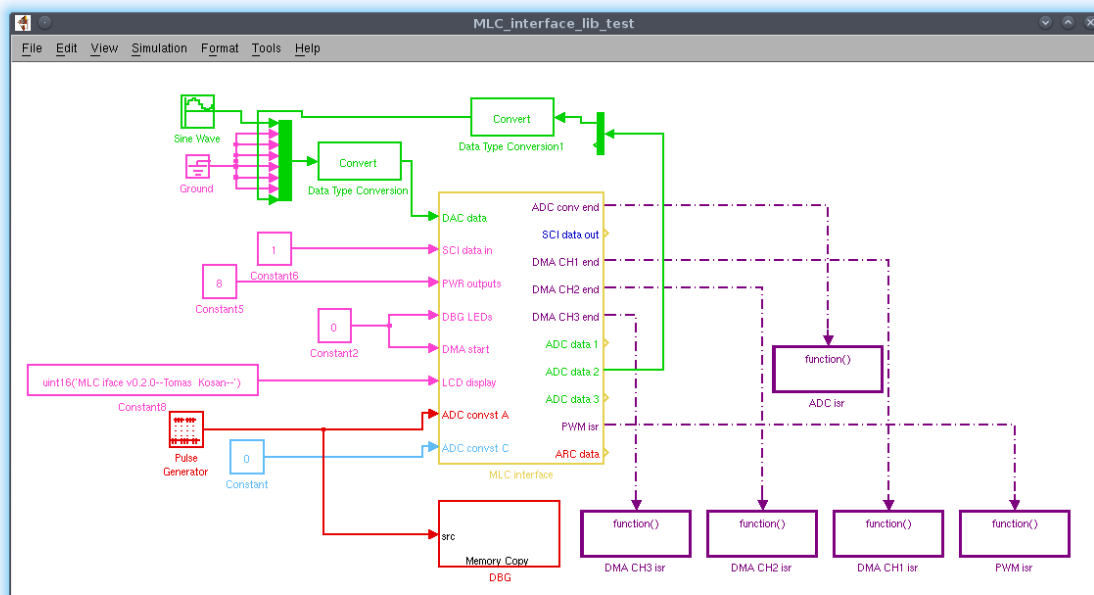
Obdobnou funkci jako *Array2DAC* má Matlab funkce *Str2Disp*. Zajišťuje periodické vypisování znaků ze vstupního pole o 32 znacích na LCD displej. Podmínkou použití tohoto vstupního signálu je nahrání základní firmware v FPGA, které zajišťuje nízkourovňovou obsluhu displeje. Tato funkce implementuje komunikační protokol entity LCD displeje z části 3.5.3 této práce.

Blok MLC interface je navržen tak, aby vygenerovaný kód umožnil rychlou implementaci regulačních schémat. Poskytuje výstupy *ADC conv end*, *DMA CH1 end*, *DMA CH2 end*, *DMA CH3 end* a *PWM isr*, ty umožňují implementovat služby přerušeni od jednotlivých událostí hardwaru MLC interface. Typicky by se výpočet regulačního algoritmu spouštěl od výstupu *DMA CH3 end*, kdy jsou již v blocích *ADC result 1,2 a 3* zapsány hodnoty naměřené A/D převodníky.

### 7.1.1. Experimentální ověření vyvinutého bloku pro Simulink

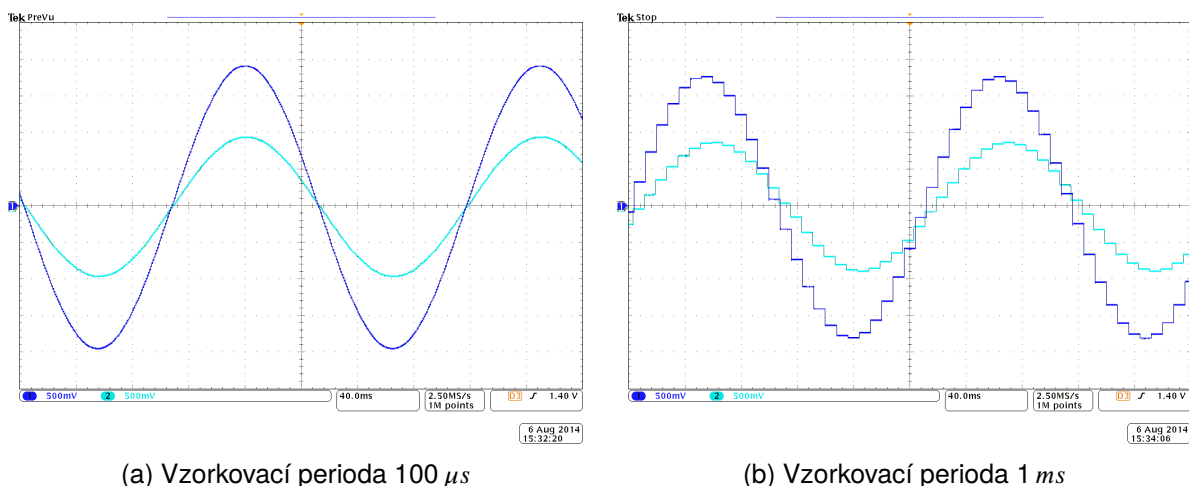
Funkčnost navrženého bloku byla ověřena na aplikaci dle Obr. 7.4. Ta ověřuje funkčnost především A/D a D/A převodníků a časování měřícího cyklu A/D převodníků. Na Obr. 7.5a jsou barevně zvýrazněny jednotlivé časové smyčky vygenerovaného kódu a na Obr. 7.5b je vyobrazena legenda. K ručnímu spouštění převodu analogově-digitálních převodníků je použit blok *Pulse Generator*, běžící ve smyčce 10  $\mu s$ . Ten generuje pulzy se střídou 1:10, tedy jeden pulz za 100  $\mu s$ .

Oscilogram pro tento způsob spouštění ukazuje Obr. 7.6a. Tento způsob spouštění převodu je vhodný pouze pro nízké vzorkovací frekvence typicky v řádu stovek mikrosekund. Zkrácením periody *Pulse Generatoru* totiž lze dosáhnout stavu, kdy mikrokontroléru nezůstane čas na ostatní úlohy (např. obsluhu hardwarových přerušeni) vlivem dodatečné režie s obsluhou generátoru pulzů a softwarovým spouštěním převodu. Vhodnější variantou je spouštění kompletně v režii hardwaru, kdy spuštění převodníků zajistí ePWM modul mikrokontroléru přes své výstupy *SOCA* nebo *SOCB* (Obr. 7.6b). Naměřené hodnoty jsou vzorkovány syn-



Obr. 7.4.: Ověřovací demo aplikace pro MLC interface v Matlab/Simulinku





Obr. 7.7.: Výsledné průběhy smyčky D/A - A/D převodníků z testovací aplikace pro různé vzorkovací frekvence smyčky,  $f=30 \text{ Hz}$ , CH1 (modrá): Výstup z D/A převodníku - kanál A (generovaná hodnota), CH2 (světle modrá): Výstup z D/A převodníku - kanál H (měřená hodnota)

oproti signálu z generátoru, protože vstupní rozsah ADC byl  $\pm 5 \text{ V}$ , zatímco DAC má výstupní rozsah  $0$  až  $5 \text{ V}$ . Na průbězích je vidět vliv nastavení vzorkovací periody této smyčky, kdy při  $100 \mu s$  máme velmi malé dopravní zpoždění mezi průběhy (Obr. 7.7a) a naopak při vzorkovací periodě  $1 ms$  je toto zpoždění znatelné a způsobuje fázový posuv mezi oběma signály (Obr. 7.7b).

Kompletní blok pro Simulink a zde zmiňovaná ukázková aplikace pro Matlab verze 2012a jsou dostupné online na [78].

## 7.2. Dílčí závěr

Byla navržena podpora pro rapid prototyping. V prostředí Matlab/Simulink byl vytvořen základní blok (model), který přidává podporu hardwarových periférií přítomných na MLC interface tak, aby bylo možno využít výhod rapid prototypingu. Správnost návrhu byla experimentálně ověřena na testovací aplikaci. Momentálně je podporován pouze mikrokontrolér TMS320F28335, do budoucna je plánováno navrhnout obdobný blok i pro mikrokontrolér TMS320F2812.

Zatím pouze teoreticky byla prověřena možnost rapid prototypingu pro hradlová pole. Je zde několik možných přístupů: např. přímý návrh v nadstavbě Simulinku Altera DSP builder a HDL coder nebo složitější, ovšem volně dostupná varianta vygenerování dílčích bloků a jejich složení dohromady za pomoci aplikace FloPoCo. Případně je možné prozkoumat i další dostupné varianty. Tato problematika bude záležitostí dalšího výzkumu.

## 8. Závěr

### 8.1. Hlavní přínos práce

Tato dizertační práce se zabývá vývojem moderního mikroprocesorového regulátoru a jeho nasazením ve složitých úlohách pro řízení vícehladinových měničů a pokročilé řízení střídavých pohonů. Byl vyvinut zcela nový regulátor, který nese označení MLC interface, a to včetně diagnostických nástrojů (zejména emulátor JTAG sběrnice s galvanickým oddělením), kompletního základního firmwaru pro hradlová pole a softwarových knihoven pro zkoumané výzkumné úlohy i nástroje pro rapid prototyping, které lze využít pro plnohodnotný model-based design v prostředí Matlab/Simulink. Navržený regulátor byl použit pro implementaci a stavbu laboratorních vzorků zkoumaných topologií vícehladinových měničů a pokročilé, extrémně výpočetně náročné algoritmy řízení a identifikace parametrů střídavých elektrických pohonů.

Vyvinutý číslicový regulátor poskytuje hardwarové prostředky pro realizaci výše zmíněných úloh, zejména umožňuje vývoj specializovaných modulátorů pro víceúrovňové topologie měničů, kdy je využito konfigurovatelnosti hradlového pole (FPGA). Využití FPGA se neomezuje jen na návrh PWM modulátorů, lze realizovat také specializované samostatné výpočetní jednotky (koprocесory) pro akceleraci časově náročných paralelních výpočtů.

JTAG emulátory jsou navrženy s ohledem na cílové použití v zarušeném prostředí laboratoře, poskytují galvanické oddělení mezi nadřazeným PC a mikroprocesorovým regulátorem a umožňují naprogramovat všechny programovatelné obvody (MCU, CPLD i FPGA) na MLC interface, tím se eliminuje počet programátorů na jeden jediný. Navržená řada JTAG emulátorů (JTAGv3, JTAGv4 a JTAGv5) se vyznačuje těmito vlastnostmi:

- Galvanické oddělení číslicového regulátoru od nadřazené jednotky (PC). Tím jsou eliminovány problémy se zemními smyčkami a pronikání rušení vznikající při provozu silové části měniče na USB sběrnici.
- Univerzalita použitého řešení umožňuje programování všech programovatelných součástek na MLC interface.
- Poslední verze JTAG emulátoru (JTAGv5) umožňuje uvést číslicový regulátor (a tím i připojený měnič) do bezpečného stavu při ztrátě komunikace přes USB sběrnici, výpadku napájení na straně PC a také při odpojení USB kabelu.

Základní softwarové vybavení vyvinuté pro MLC interface poskytuje uživateli abstraktní vrstvu mezi hardwarem a implementovanými algoritmy, uživatel se tedy může soustředit na samotný návrh a implementaci regulačního algoritmu bez podrobné znalosti hardwaru číslicového regulátoru MLC interface. Základní software pro FPGA pak zjednodušuje uživateli implementaci specializovaných entit (PWM modulátorů atp.), kdy uživatel definuje pouze registry entity mapované do paměti mikrokontroléru a veškeré nízkoúrovňové úlohy nutné pro správnou funkci jsou již v základním kódu zahrnuty.

MLC interface byl nasazen autorem při výzkumu řízení nepřímých měničů kmitočtu napěťových typů s topologiemi:

- Čtyřúrovňový kompletní napěťový nepřímý měnič kmitočtu s plovoucími kondenzátory (4L-FLC).
  - Byl navržen komplexní PD-PWM modulátor s balanční tabulkou, který umožňuje regulaci napětí na plovoucích kondenzátorech bez zásahu mikrokontroléru.
  - Byla navržena univerzální přednabíjecí procedura měniče, použitelná pro jakýkoli počet úrovní měniče a zároveň je aplikovatelná na měnič v režimu střídače i usměrňovače.
  - Návrh regulačních algoritmů a modulátoru byl experimentálně ověřen na prototypu o jmenovitém výkonu 35 kVA.
- Tříúrovňový kompletní napěťový nepřímý měnič kmitočtu s upínacími diodami (3L-NPC).
  - Byl navržen modulární PWM modulátor se synchronními výstupy.
  - Návrh regulačních algoritmů a modulátoru byl experimentálně ověřen na prototypu o jmenovitém výkonu 35 kVA.
- Jednofázový sedmiúrovňový napěťový pulzní usměrňovač se sériově řazenými H-můstky (7L-CHB).
  - Byl navržen blok PWM modulátoru pro jednu buňku H-můstku s nastavitelným přesazením referenční pily a generátorem mrtvých časů.
  - Pro synchronizaci napěťového pulzního usměrňovače s napájecí sítí byl použit algoritmus SOGI, vylepšený o možnost přeladění. Rychlost a správnost zavěšení SOGI byla vylepšena pomocí startovací podmínky tohoto algoritmu.
  - Byl navržen algoritmus pro vyvažování napětí meziobvodu jednotlivých sériově řazených buněk využívající PI regulátory a změnu amplitudy modulačního signálu pro jednotlivé buňky.



- Návrh regulačních algoritmů, synchronizace, vyvažování napětí meziobvodů a modulátoru byl experimentálně ověřen na postaveném prototypu měniče při výkonu 2 kVA.

Hardware tedy umožnil vyvinout kompletní programové vybavení pro tyto měniče včetně jejich startovacích a vypínacích procedur, modulátorů, řízení atd. Byly provedeny experimenty s různými algoritmy řízení a především byly odladěny algoritmy modulátorů s balancováním napětí na kondenzátorech. Bylo vyzkoušeno několik různých přístupů k návrhu modulátoru do hradlového pole, od nejjednodušších čítačů s komparačními registry až po modulátor zajišťující aktivní balancování napětí na kondenzátorech víceúrovňového měniče. Na těchto experimentech se také ověřila dostatečná odolnost vyvinutého hardwaru proti rušením vznikajícím při provozu silové části měniče, což je ukázáno výsledky experimentů.

Dále byly zkoumány, navrženy a s využitím nového regulátoru implementovány pokročilé algoritmy bezsensorového řízení střídavých elektrických pohonů. Konkrétně v této práci je prezentováno první nasazení negausovského stochastického filtru v elektrických pohonech, přesněji - bylo navrženo řízení střídavých elektrických strojů s využitím estimátoru rychlosti a polohy rotoru s využitím marginalizovaného částicového filtru, který kombinuje techniku částicové filtrace s Kalmanovým filtrem, který tvoří každou částici (particle).

Navržené řízení bylo experimentálně odzkoušeno na postaveném modelu pohonu s PMSM o výkonu 10,7 kW. Hlavní výsledky tohoto výzkumu byly publikovány v [77], který byl na prestižním sympoziu Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE oceněn „best session paper awards“.

## 8.2. Perspektivní směry dalšího výzkumu

Vyvinutá vývojová platforma umožňuje řízení měničů až o sedmi úrovních, nabízí se realizace sedmiúrovňového měniče s plovoucími kondenzátory. Obecně u měničů s plovoucími kondenzátory je vhodné výzkum zaměřit na možnost redukce počtu čidel pro měření napětí na plovoucích kondenzátorech pomocí měření výstupního fázového napětí měniče v kombinaci se známou aktuální spínací kombinací.

Široké pole k prozkoumání se nabízí v oblasti prediktivních modulátorů pro měniče s plovoucími kondenzátory, kdy by se předpočítaná balancovací tabulka nahradila prediktivním regulátorem, který by zajišťoval správné úrovně napětí na příslušných kondenzátorech.

Měnič založený na topologii H můstků byl prozatím realizován pouze v jednofázové verzi, dalším krokem bude vícefázová varianta a návrh vektorového modulátoru a vektorové regulace motoru a pulzního usměrňovače.

Vyzkoušená řízení komplexních víceúrovňových topologií měničů také ukázala, že stávající mikrokontrolérová platforma (TMS320F28335) se blíží ke svému limitu výpočetního výkonu. Dále je tedy vhodné zaměřit pozornost na vývoj mikrokontrolérového modulu

s vyšším výkonem a případně i více nezávislými jádry, jako slibný se jeví mikrokontrolér TMS320F28377.

V oblasti rapid prototypingu budou dále zkoumány možnosti usnadnění vývoje regulačních algoritmů přímo pro hradlová pole. Již hotový blok MLC interface pro Matlab bude vylepšen o možnost jeho simulace, včetně všech časových prodlev daných hardwarovými součástmi MLC interface.

## Literatura

- [1] LAI, Jih-Sheng, PENG, Fang Zheng. *Multilevel converters-a new breed of power converters*. Industry Applications, IEEE Transactions on. 1996, roč. 32, č. 3, s. 509–517. Dostupný z WWW: <http://dx.doi.org/10.1109/28.502161>. ISSN 0093-9994.
- [2] RODRIGUEZ, J., LAI, Jih-Sheng, PENG, Fang Zheng. *Multilevel inverters: a survey of topologies, controls, and applications*. Industrial Electronics, IEEE Transactions on. 2002, roč. 49, č. 4, s. 724–738. Dostupný z WWW: <http://dx.doi.org/10.1109/TIE.2002.801052>. ISSN 0278-0046.
- [3] RODRIGUEZ, J., FRANQUELO, L.G., KOURO, S. et al. *Multilevel Converters: An Enabling Technology for High-Power Applications*. Proceedings of the IEEE. 2009, roč. 97, č. 11, s. 1786–1817. Dostupný z WWW: <http://dx.doi.org/10.1109/JPROC.2009.2030235>. ISSN 0018-9219.
- [4] PENG, Fang Zheng. *A generalized multilevel inverter topology with self voltage balancing*. Industry Applications, IEEE Transactions on. 2001, roč. 37, č. 2, s. 611–618. Dostupný z WWW: <http://dx.doi.org/10.1109/28.913728>. ISSN 0093-9994.
- [5] CHEN, Alian, HE, Xiangning. *Research on Hybrid-Clamped Multilevel-Inverter Topologies*. Industrial Electronics, IEEE Transactions on. 2006, roč. 53, č. 6, s. 1898–1907. Dostupný z WWW: <http://dx.doi.org/10.1109/TIE.2006.885154>. ISSN 0278-0046.
- [6] SHUKLA, A., GHOSH, A., JOSHI, A. *Improved Multilevel Hysteresis Current Regulation and Capacitor Voltage Balancing Schemes for Flying Capacitor Multilevel Inverter*. Power Electronics, IEEE Transactions on. 2008, roč. 23, č. 2, s. 518–529. Dostupný z WWW: <http://dx.doi.org/10.1109/TPEL.2007.915788>. ISSN 0885-8993.
- [7] MCGRATH, B.P., HOLMES, D.G. *Natural Capacitor Voltage Balancing for a Flying Capacitor Converter Induction Motor Drive*. Power Electronics, IEEE Transactions on. 2009, roč. 24, č. 6, s. 1554–1561. Dostupný z WWW: <http://dx.doi.org/10.1109/TPEL.2009.2016567>. ISSN 0885-8993.

- 
- [8] WILKINSON, R.H., MEYNARD, T.A., MOUTON, H. Du Toit. *Natural Balance of Multi-cell Converters: The Two-Cell Case*. Power Electronics, IEEE Transactions on. 2006, roč. 21, č. 6, s. 1649–1657. Dostupný z WWW: (<http://dx.doi.org/10.1109/TPEL.2006.882958>). ISSN 0885-8993.
- [9] WILKINSON, R.H., MEYNARD, T.A., MOUTON, H. Du Toit. *Natural Balance of Multi-cell Converters: The General Case*. Power Electronics, IEEE Transactions on. 2006, roč. 21, č. 6, s. 1658–1666. Dostupný z WWW: (<http://dx.doi.org/10.1109/TPEL.2006.882951>). ISSN 0885-8993.
- [10] KOKEŠ, P. *Balancování napětí ve vícehladinovém střídači s plovoucími kondenzátory*. Praha : Akademie věd ČR, 2007. výzkumná zpráva č.: Z-1415/07.
- [11] VALOUCH, V., KOKEŠ, P., ŠKRAMLÍK, J. *Vícehladinové napěťové měniče s plovoucími kondenzátory na síti a jejich řízení*. Praha : Akademie věd ČR, 2008. výzkumná zpráva č.: Z-1418/08.
- [12] ALTERA CORPORATION. *All Development Kits*. [cit. 10.1.2013]. Dostupný z WWW: ([http://www.altera.com/products/devkits/kit-dev\\_platforms.jsp](http://www.altera.com/products/devkits/kit-dev_platforms.jsp)).
- [13] XILINX INC. *Spartan-6 Boards and Kits*. [cit. 10.1.2013]. Dostupný z WWW: ([http://www.xilinx.com/products/boards\\_kits/spartan6.htm](http://www.xilinx.com/products/boards_kits/spartan6.htm)).
- [14] TEXAS INSTRUMENTS INC. *Kits & Evaluation Modules for C2000 Microcontrollers*. [cit. 10.1.2013]. Dostupný z WWW: ([http://www.ti.com/lstds/ti/microcontroller/32-bit\\_c2000/tools\\_software.page#kits](http://www.ti.com/lstds/ti/microcontroller/32-bit_c2000/tools_software.page#kits)).
- [15] STMICROELECTRONICS. *STEVAL-IHM039V1 Dual motor drive control stage based on the STM32F415ZG microcontroller*. [cit. 9.1.2013]. Dostupný z WWW: (<http://www.st.com/web/catalog/tools/FM116/SC1077/PF253396>).
- [16] FREESCALE SEMICONDUCTOR, INC. *Automotive Motor Control Development Solutions*. [cit. 9.1.2013]. Dostupný z WWW: (<http://www.freescale.com/webapp/sps/site/overview.jsp?code=AUTOMCDEVKITS>).
- [17] XILINX INC. *Avnet Spartan-6 / Intel Atom Development Kit*. [cit. 18.1.2013]. Dostupný z WWW: (<http://www.xilinx.com/products/boards-and-kits/AES-S6NITX-LX75T.htm>).
- [18] KATEDRA KYBERNETIKY, AUTOMATICKÉ ŘÍZENÍ. *Úvodní stránka AŘ*. [cit. 27.1.2013]. Dostupný z WWW: (<http://www.kky.zcu.cz/cs/research-fields/rex-control-system>).
-

- 
- [19] BIELEWICZ, Z., DEBOWSKI, L., LOWIEC, E. *A DSP and FPGA based integrated controller development solutions for high performance electric drives*. In *Industrial Electronics, 1996. ISIE '96., Proceedings of the IEEE International Symposium on. 1996*, s. 679–684 vol.2. Dostupný z WWW: <http://dx.doi.org/10.1109/ISIE.1996.551024>).
- [20] BUENO, E.J., HERNANDEZ, A., RODRIGUEZ, F.J. et al. *A DSP- and FPGA-Based Industrial Control With High-Speed Communication Interfaces for Grid Converters Applied to Distributed Power Generation Systems*. *Industrial Electronics, IEEE Transactions on. 2009*, roč. 56, č. 3, s. 654–669. Dostupný z WWW: <http://dx.doi.org/10.1109/TIE.2008.2007043>). ISSN 0278-0046.
- [21] GIRÓN, C., RODRIGUEZ, F.J., HUERTA, F., BUENO, E. *Implementing high speed communication buses for a FPGA-DSP architecture for digital control of power electronics*. In *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on. 2007*, s. 1–6. Dostupný z WWW: <http://dx.doi.org/10.1109/WISP.2007.4447601>).
- [22] HERNÁNDEZ, A., MATEOS, R., RODRÍGUEZ, F.J. et al. *Using FPGA as coprocessor device in the controllers of grid power converters*. In *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE. 2008*, s. 2405–2410. Dostupný z WWW: <http://dx.doi.org/10.1109/IECON.2008.4758333>).
- [23] ATALIK, T., DENIZ, M., KOC, E. et al. *Multi-DSP and -FPGA-Based Fully Digital Control System for Cascaded Multilevel Converters Used in FACTS Applications*. *Industrial Informatics, IEEE Transactions on. 2012*, roč. 8, č. 3, s. 511–527. Dostupný z WWW: <http://dx.doi.org/10.1109/TII.2012.2194160>). ISSN 1551-3203.
- [24] BLERIS, L.G., VOUZIS, P.D., ARNOLD, M.G., KOTHARE, M.V. *A co-processor FPGA platform for the implementation of real-time model predictive control*. In *American Control Conference, 2006. 2006*. Dostupný z WWW: <http://dx.doi.org/10.1109/ACC.2006.1656499>).
- [25] SCHON, T., GUSTAFSSON, F., NORDLUND, P.-J. *Marginalized particle filters for mixed linear/nonlinear state-space models*. *Signal Processing, IEEE Transactions on. 2005*, roč. 53, č. 7, s. 2279–2289. Dostupný z WWW: <http://dx.doi.org/10.1109/TSP.2005.849151>). ISSN 1053-587X.
- [26] SAHA, S., OZKAN, E., GUSTAFSSON, F., ŠMÍDL, V. *Marginalized particle filters for Bayesian estimation of Gaussian noise parameters*. In *Information Fusion (FUSION), 2010 13th Conference on. 2010*, s. 1–8.
-

- 
- [27] OZKAN, E., SAHA, S., GUSTAFSSON, F., SMIDL, V. *Non-parametric bayesian measurement noise density estimation in non-linear filtering*. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. 2011, s. 5924–5927. Dostupný z WWW: <http://dx.doi.org/10.1109/ICASSP.2011.5947710>).
- [28] THIELEMANS, S., VYNCKE, T.J., JACXSENS, M., MELKEBEEK, J.A. *FPGA implementation of online finite-set model based predictive control for power electronics*. In *Predictive Control of Electrical Drives and Power Electronics (PRECEDE), 2011 Workshop on*. 2011, s. 63–69. Dostupný z WWW: <http://dx.doi.org/10.1109/PRECEDE.2011.6079085>).
- [29] LEUER, M., BOCKER, J. *Fast online model predictive control of IPMSM using parallel computing on FPGA*. In *Electric Machines Drives Conference (IEMDC), 2013 IEEE International*. 2013, s. 1017–1022. Dostupný z WWW: <http://dx.doi.org/10.1109/IEMDC.2013.6556221>).
- [30] THIELEMANS, S., VYNCKE, T.J., JACXSENS, M., MELKEBEEK, J.A. *FPGA implementation of online finite-set model based predictive control for power electronics*. In *Predictive Control of Electrical Drives and Power Electronics (PRECEDE), 2011 Workshop on*. 2011, s. 63–69. Dostupný z WWW: <http://dx.doi.org/10.1109/PRECEDE.2011.6079085>).
- [31] XU, Fang, CHEN, Hong, JIN, Weiwei, XU, Yueting. *FPGA implementation of nonlinear model predictive control*. In *Control and Decision Conference (2014 CCDC), The 26th Chinese*. 2014, s. 108–113. Dostupný z WWW: <http://dx.doi.org/10.1109/CCDC.2014.6852127>).
- [32] WAN, E.A, VAN DER MERWE, R. *The unscented Kalman filter for nonlinear estimation*. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. 2000, s. 153–158. Dostupný z WWW: <http://dx.doi.org/10.1109/ASSPCC.2000.882463>).
- [33] LIANG-QUN, Li, HONG-BING, Ji, JUN-HUI, Luo. *The iterated extended Kalman particle filter*. In *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on*. 2005, s. 1213–1216. Dostupný z WWW: <http://dx.doi.org/10.1109/ISCIT.2005.1567087>).
- [34] GUO-HUI, Li, YA-AN, Li, HONG, Yang, LIN, Cui. *Improved unscented Kalman particle filter*. In *Mechatronics and Automation (ICMA), 2010 International Conference on*. 2010, s. 804–808. Dostupný z WWW: <http://dx.doi.org/10.1109/ICMA.2010.5589030>).
-

- [35] SMIDL, V., PEROUTKA, Z. *Marginalized particle filter for sensorless control of PMSM drives*. In IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society. 2012, s. 1877–1882. Dostupný z WWW: (<http://dx.doi.org/10.1109/IECON.2012.6388915>).
- [36] MOLNÁR, J. *Jednofázový proudový pulzní usměrňovač*. Plzeň : Západočeská univerzita v Plzni, 2007. Dizertační práce.
- [37] TEXAS INSTRUMENTS INC. *Comparing Bus Solutions*. [cit. 14.3.2013]. Dostupný z WWW: (<http://www.ti.com/lit/an/s11a067b/s11a067b.pdf>).
- [38] TEXAS INSTRUMENTS INC. *Low-Voltage Differential Signaling (LVDS)*. [cit. 14.3.2013]. Dostupný z WWW: (<http://www.ti.com/lit/an/s11a014a/s11a014a.pdf>).
- [39] KOŠAN, T., MOLNÁR, J., STREIT, L., POLÁČEK, L., PEROUTKA, Z. *Complete Design of Down-Scale Prototype of Mining Machine Converter based on Four-Level Voltage-Source Converter with Flying Capacitors*. In EPE PEMC 2012 ECCE Europe - 15th International Power Electronics and Motion Conference and Exposition. New York : IEEE, 2012, s. ISBN 978-1-4673-1971-3.
- [40] ALTERA CORPORATION. *Nios II Processor: The World's Most Versatile Embedded Processor*. [cit. 20.2.2014]. Dostupný z WWW: (<http://www.altera.com/devices/processor/nios2/ni2-index.html>).
- [41] XILINX INC. *MicroBlaze Soft Processor Core*. [cit. 20.2.2014]. Dostupný z WWW: (<http://www.xilinx.com/tools/microblaze.htm>).
- [42] ALTERA CORPORATION. *SoC Overview*. [cit. 20.2.2014]. Dostupný z WWW: (<http://www.altera.com/devices/processor/soc-fpga/overview/proc-soc-fpga.html>).
- [43] XILINX INC. *Zynq-7000 All Programmable SoC*. [cit. 20.2.2014]. Dostupný z WWW: (<http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/index.htm>).
- [44] TEXAS INSTRUMENTS INC. *XDS100 - Texas Instruments Embedded Processors Wiki*. [cit. 8.12.2012]. Dostupný z WWW: (<http://processors.wiki.ti.com/index.php/XDS100>).
- [45] ANALOG DEVICES INC. *ADUM5000: Isolated DC/DC Converter*. [cit. 2.10.2012]. Dostupný z WWW: (<http://www.analog.com/en/interface-isolation/digital-isolators/adum5000/products/product.html>).
- [46] ANALOG DEVICES INC. *ADUM3160: Full/Low Speed USB Digital Isolator*. [cit. 2.10.2012]. Dostupný z WWW: (<http://www.analog.com/en/interface-isolation/digital-isolators/adum3160/products/product.html>).

- 
- [47] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD. *FT2232D Dual USB to Serial UART/FIFO IC*. [cit. 20.12.2012]. Dostupný z WWW: ([http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS%5C\\_FT2232D.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS%5C_FT2232D.pdf)).
- [48] ANALOG DEVICES INC. *Dual-Channel Digital Isolators ADuM1200/ADuM1201*. [cit. 2.10.2012]. Dostupný z WWW: ([http://www.analog.com/static/imported-files/data%5C\\_sheets/ADuM1200%5C\\_1201.pdf](http://www.analog.com/static/imported-files/data%5C_sheets/ADuM1200%5C_1201.pdf)).
- [49] SPECTRUM DIGITAL INC. *eZdsp™ F28335 board with Code Composer Studio DSK tools and power supply*. [cit. 14.3.2014]. Dostupný z WWW: ([http://www.spectrumdigital.com/product\\_info.php?products\\_id=198](http://www.spectrumdigital.com/product_info.php?products_id=198)).
- [50] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD. *FT2232H Dual High Speed USB to Multipurpose UART/FIFO IC*. [cit. 2.1.2013]. Dostupný z WWW: ([http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT2232H.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT2232H.pdf)).
- [51] WASCHK, K. *Universal JTAG library, server and tools*. [cit. 15.1.2013]. Dostupný z WWW: (<http://urjtag.org>).
- [52] TEXAS INSTRUMENTS INC., ASSET INTERTECH INC. *Serial Vector Specification*. [cit. 14.3.2013]. Dostupný z WWW: ([http://www.jtagtest.com/pdf/svf\\_specification.pdf](http://www.jtagtest.com/pdf/svf_specification.pdf)).
- [53] ALTERA CORPORATION. *Cyclone III FPGA Family Overview*. [cit. 20.3.2014]. Dostupný z WWW: (<http://www.altera.com/devices/fpga/cyclone3/overview/cy3-overview.html>).
- [54] XILINX, INC. *XC95144XL High Performance CPLD*. [cit. 20.3.2014]. Dostupný z WWW: ([http://www.xilinx.com/support/documentation/data\\_sheets/ds056.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds056.pdf)).
- [55] ANALOG DEVICES, INC. *AD5328: 2.5 V to 5.5 V Octal Voltage Output 12-Bit DACs in 16-Lead TSSOP*. [cit. 20.3.2014]. Dostupný z WWW: (<http://www.analog.com/en/digital-to-analog-converters/dac-converters/ad5328/products/product.html>).
- [56] ANALOG DEVICES, INC. *AD7609: 8-Channel Differential DAS with 18-Bit, Bipolar, Simultaneous Sampling ADC*. [cit. 20.3.2014]. Dostupný z WWW: (<http://www.analog.com/en/analog-to-digital-converters/ad-converters/ad7609/products/product.html>).
- [57] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD. *FT232R - USB UART IC*. [cit. 2.3.2013]. Dostupný z WWW: ([http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf)).
-



- [58] TEXAS INSTRUMENTS INC. *TMS320C6000 DSP Designing for JTAG Emulation Reference Guide*. [cit. 14.3.2014]. Dostupný z WWW: (<http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=SPRU641&fileType=pdf>).
- [59] *Doxygen : Main Page*. [cit. 14.3.2014]. Dostupný z WWW: (<http://www.stack.nl/~dimitri/doxygen>).
- [60] KOŠAN, T. *MLC interface driver*. [cit. 20.6.2014]. Dostupný z WWW: ([http://home.zcu.cz/~kosan/?Dokumentace\\_MLC\\_driver](http://home.zcu.cz/~kosan/?Dokumentace_MLC_driver)).
- [61] KOŠAN, T., BEDNÁŘ, B. *Mikrokontrolérový modul TMS320F2812 pro MLC interface*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22160-006-2013.
- [62] KOŠAN, T. *Mikrokontrolérový modul TMS320F28335 pro MLC interface*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22160-002-2013.
- [63] KOŠAN, T. *Mikrokontrolérový modul TMS570LS3137 pro MLC interface*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22160-003-2013.
- [64] ANALOG DEVICES, INC. *ADM3053: Signal and Power Isolated CAN Transceiver with Integrated Isolated DC-to-DC Converter*. [cit. 20.3.2014]. Dostupný z WWW: (<http://www.analog.com/en/interface-isolation/can/adm3053/products/product.html>).
- [65] MCGRATH, B.P., MEYNARD, T., GATEAU, G., HOLMES, D.G. *Optimal Modulation of Flying Capacitor and Stacked Multicell Converters Using a State Machine Decoder*. Power Electronics, IEEE Transactions on. 2007, roč. 22, č. 2, s. 508–516. Dostupný z WWW: (<http://dx.doi.org/10.1109/TPEL.2006.889932>). ISSN 0885-8993.
- [66] SHUKLA, A., GHOSH, A., JOSHI, A. *Natural Balancing of Flying Capacitor Voltages in Multicell Inverter Under PD Carrier-Based PWM*. Power Electronics, IEEE Transactions on. 2011, roč. 26, č. 6, s. 1682–1693. Dostupný z WWW: (<http://dx.doi.org/10.1109/TPEL.2010.2089807>). ISSN 0885-8993.
- [67] XILINX, INC. *XC9572 In-System Programmable CPLD*. [cit. 20.3.2014]. Dostupný z WWW: ([http://www.xilinx.com/support/documentation/data\\_sheets/ds065.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds065.pdf)).
- [68] PEROUTKA, Z., BLAHNÍK, V. *Algoritmy řízení a regulace třífázového NPU pro těžní stroj: základní algoritmy regulace*. Plzeň : Západočeská univerzita v Plzni, 2010. výzkumná zpráva č.: 22190-03-2010.
- [69] KAMENICKÝ, P., GLASBERGER, V. *Implementace algoritmů řízení tříúrovňového měniče s upínacími diodami*. Plzeň : Západočeská univerzita v Plzni, 2012. výzkumná zpráva č.: 22190-059-2012.

- [70] KOŠAN, T. *Implementace algoritmu řízení tyristorového usměrňovače do mikropočítače*. Plzeň : Západočeská univerzita v Plzni, 2008. Diplomová práce.
- [71] SEPAHVAND, H., KHAZRAEI, M., FERDOWSI, M., CORZINE, K.A. *Capacitor voltage regulation and pre-charge routine for a flying capacitor active rectifier*. In Energy Conversion Congress and Exposition (ECCE), 2012 IEEE. 2012, s. 4107–4112. Dostupný z WWW: (<http://dx.doi.org/10.1109/ECCE.2012.6342265>).
- [72] KAMENICKÝ, P., JANÍK, D., KOŠAN, T., PEROUTKA, Z. *Vektorové řízení asynchronního motoru s měničem 4L-FLC*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22190-055-2013.
- [73] JANIK, D., TALLA, J., KOMRSKA, T., PEROUTKA, Z. *Optimization of SOGI PLL for single-phase converter control systems: Second order generalized integrator (SOGI)*. In Applied Electronics (AE), 2013 International Conference on. 2013, s. 1–4.
- [74] BLAHNÍK, V. *Algoritmy řízení a regulace sestavy trakčního měniče se středofrekvenčním transformátorem*. Plzeň : Západočeská univerzita v Plzni, 2011. Dizertační práce.
- [75] ALTERA CORPORATION. *DSP builder*. [cit. 10.1.2013]. Dostupný z WWW: (<http://www.altera.com/products/software/products/dsp/dsp-builder.html>).
- [76] DINECHIN, F. de, PASCA, B. *Designing Custom Arithmetic Data Paths with FloPoCo*. Design Test of Computers, IEEE. 2011, roč. 28, č. 4, s. 18–27. Dostupný z WWW: (<http://dx.doi.org/10.1109/MDT.2011.44>). ISSN 0740-7475.
- [77] SMIDL, V., NEDVED, R., KOSAN, T., PEROUTKA, Z. *FPGA implementation of marginalized particle filter for sensorless control of PMSM drives*. In Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE. 2013, s. 8227–8232. Dostupný z WWW: (<http://dx.doi.org/10.1109/IECON.2013.6700510>).
- [78] KOŠAN, T. *Model MLC interface v MATLAB-Simulinku*. [cit. 14.7.2014]. Dostupný z WWW: ([http://home.zcu.cz/~kosan/?Model\\_MLC\\_interface\\_v\\_MATLAB-Simulinku](http://home.zcu.cz/~kosan/?Model_MLC_interface_v_MATLAB-Simulinku)).

## Seznam autorových publikací

### Seznam autorových publikací prezentovaných na mezinárodních konferencích (chronologicky)

- [A1] KOŠAN, T. *Basic entities for motor control implemented in VHDL*. In 2011 International Conference on Applied Electronics. Plzeň : Západočeská univerzita v Plzni, 2011, s. 197–200. ISBN 978-80-7043-987-6.
- [A2] KOŠAN, T., MOLNÁR, J., STREIT, L., POLÁČEK, L., PEROUTKA, Z. *Complete Design of Down-Scale Prototype of Mining Machine Converter based on Four-Level Voltage-Source Converter with Flying Capacitors*. In EPE PEMC 2012 ECCE Europe - 15th International Power Electronics and Motion Conference and Exposition. New York : IEEE, 2012, s. ISBN 978-1-4673-1971-3.
- [A3] KOŠAN, T., PEROUTKA, Z., GLASBERGER, T. *Multi-level Converter Interface - A New Hardware Platform For Multi-level Converters Development*. In 2012 International Conference on Applied Electronics. Plzeň : Západočeská univerzita v Plzni, 2012, s. 143–147. ISBN 978-80-261-0038-6.
- [A4] BEDNÁŘ, B., STREIT, L., KOŠAN, T. *Diagnostic tool for Lithium and Lead-Acid Battery*. In 4th International Conference on Clean Electrical Power Renewable Energy Resources Impact. New York : IEEE, 2013, s. 84–86. ISBN 978-1-4673-4430-2.
- [A5] JANÍK, D., KOŠAN, T., GLASBERGER, T., PEROUTKA, Z. *Comparison of control solutions for control of four-level converter with flying capacitors*. In 2013 International Conference on Applied Electronics. Pilsen : University of West Bohemia, 2013, s. 117–120. ISBN 978-80-261-0166-6.
- [A6] JANÍK, D., KOŠAN, T., KAMENICKÝ, P., PEROUTKA, Z. *Universal Precharging Method for dc-Link and Flying Capacitors of Four-Level Flying Capacitor Converter*. In Proceedings of the IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society. Vienna : IEEE, 2013, s. 6320–6325. ISBN 978-1-4799-0223-1.

- [A7] KOŠAN, T., ZEMAN, M., STREIT, L., PEROUTKA, Z., JANÍK, D. *Active Voltage Balancing Control with Phase Disposition PWM for 4-level Flying Capacitor Converter*. In EPE 13 ECCE Europe. Brussel : EPE Association, 2013, s. P.1–P.8. ISBN 978-90-75815-17-7.
- [A8] ŠMÍDL, V., NEDVĚD, R., KOŠAN, T., PEROUTKA, Z. *FPGA Implementation of Marginalized Particle Filter for Sensorless Control of PMSM Drives*. In Proceedings of the IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society. Vienna : IEEE, 2013, s. 8219–8224. ISBN 978-1-4799-0223-1.
- [A9] JANÍK, D., KOŠAN, T., BLAHNÍK, V. et al. *Complete solution of 4-level Flying Capacitor Converter for Medium-Voltage Drives with Active Voltage Balancing Control with Phase-Disposition PWM*. In the 16th Conference on Power Electronics and Applications, EPE'14-ECCE Europe. 2014.
- [A10] JANÍK, D., KOŠAN, T., SADSKÝ, J. *Implementation of SVPWM algorithm without trigonometric functions*. In 2014 International Conference on Applied Electronics. 2014, s. 131–134. ISBN 978-80-261-0276-2.
- [A11] ŠMÍDL, V., NEDVĚD, R., KOŠAN, T., PEROUTKA, Z. *FPGA Implementation of Rao-Blackwellized Particle Filter and its Application to Sensorless Drive Control*. In IFAC 2014 - 19th World Congress of the International Federation of Automatic Control. 2014, s. 8669–8674.

## **Seznam autorových publikací prezentovaných na českých konferencích (chronologicky)**

- [A12] KOŠAN, T. *Vývoj JTAG emulátoru pro regulátory s DSP Texas Instruments C2000*. In Elektrotechnika a informatika 2009. Část 2., Elektronika. Plzeň : Západočeská univerzita, 2009, s. 75–79. ISBN 978-80-7043-809-1.
- [A13] KOŠAN, T. *Nízkoúrovňová řídicí jednotka pro výkonovou část čtyřúrovňového měniče s plovoucími kondenzátory implementovaná do CPLD*. In Elektrotechnika a informatika 2011. Část 2. Elektronika. Plzeň : Západočeská univerzita v Plzni, 2011, s. 55–58. ISBN 978-80-261-0015-7.
- [A14] KOŠAN, T. *Univerzální galvanicky oddělený JTAG emulátor pro procesory Texas Instruments s podporou programování CPLD a FPGA*. In Elektrotechnika a informatika 2012. Část 2., Elektronika. Plzeň : Západočeská univerzita v Plzni, 2012, s. 71–74. ISBN 978-80-261-0119-2.

- [A15] BEDNÁŘ, B., STREIT, L., KOŠAN, T. *Diagnostický nástroj pro Li-Ion a olověné baterie*. In *Elektrotechnika a informatika 2013. Část 2., Elektronika*. Plzeň : Západočeská univerzita, 2013, s. 5–8. ISBN 978-80-261-0232-8.
- [A16] JANÍK, D., PEROUTKA, Z., KOŠAN, T. *Řídicí platforma pro víceúrovňové měniče s FPGA a DSP: koncepce řízení 4L-FLC měniče*. In *Elektrické pohony : XXXIII. konference*. Praha : Česká elektrotechnická společnost, 2013, s. 1–6. ISBN 978-80-02-02457-6.
- [A17] STREIT, L., DRÁBEK, P., KOŠAN, T. *Simulace tramvaje se systémem akumulace energie*. In *Elektrické pohony : XXXIII. Konference*. Praha : Česká elektrotechnická společnost, 2013, s. 1–8. ISBN 978-80-02-02457-6.

### Seznam autorových funkčních vzorků (chronologicky)

- [A18] MOLNÁR, J., KOŠAN, T. *Aktivní filtr PFC na principu měniče step-down*, 2009.
- [A19] MOLNÁR, J., KOŠAN, T. *Budič tyristorového třífázového můstkového usměrňovače s řídicí a zobrazovací jednotkou*, 2009.
- [A20] KOŠAN, T. *JTAG adaptér pro DSP Texas Instruments*, 2010.
- [A21] KOŠAN, T. *Modulární regulátor pohonu s asynchronním motorem na bázi FPGA Altera Cyclone II*, 2010.
- [A22] MICHALÍK, J., MOLNÁR, J., KOŠAN, T. *Redukce pro připojení modulu SKiiP k číslicovému regulátoru*, 2010.
- [A23] MICHALÍK, J., MOLNÁR, J., KOŠAN, T. *Univerzální budič IGBT pro čtyři větve střídače*, 2010.
- [A24] MICHALÍK, J., MOLNÁR, J., PEROUTKA, Z., KOŠAN, T. *Regulátor jednofázového proudového pulzního usměrňovače založený na regulátoru fázového posunu*, 2010.
- [A25] MICHALÍK, J., MOLNÁR, J., PEROUTKA, Z., KOŠAN, T. *Regulátor jednofázového proudového pulzního usměrňovače založený na regulátoru fázového posunu s korekcí*, 2010.
- [A26] MOLNÁR, J., KOŠAN, T. *Třífázový můstkový tyristorový usměrňovač*, 2010.
- [A27] KOŠAN, T. *Modul budiče čtyřúrovňového střídače s plovoucími kondenzátory založený na CPLD*, 2011.
- [A28] KOŠAN, T. *Transparentní USB izolátor odolný proti EMI*, 2011.
- [A29] KOŠAN, T. *Univerzální izolovaný JTAG adaptér verze 4*, 2011.

- [A30] MOLNÁR, J., KOŠAN, T. *Testovací modul A/D převodníku mapovaného do adresního prostoru pro TMS320F2812*, 2011.
- [A31] MOLNÁR, J., KOŠAN, T., KRÁL, V., STREIT, L. *Měnič Měnič 4L-FLC 20kW 20kW*, 2011.
- [A32] KOŠAN, T. *JTAG emulátor s galvanickým oddělením, seriovou linkou a fail-safe funkcí*, 2012.
- [A33] KOŠAN, T. *MLC interface - Interface pro víceúrovňové měniče a výpočetně náročné algoritmy řízení*, 2012.
- [A34] KOŠAN, T., POLÁČEK, L. *Hlavní řídicí karta systému REMCS - MCU01*, 2012.
- [A35] MOLNÁR, J., KOŠAN, T., STREIT, L. *Tříhladinový NPC napěťový střídač*, 2012.
- [A36] MOLNÁR, J., STREIT, L., KOŠAN, T. *Tříhladinový NPC napěťový pulzní usměrňovač*, 2012.
- [A37] KOŠAN, T. *Mikrokontrolerový modul pro vývojový kit MLC interface s TMS320F28335*, 2013.
- [A38] KOŠAN, T. *Mikrokontrolerový modul pro vývojový kit MLC interface s TMS570LS3137*, 2013.
- [A39] KOŠAN, T., BEDNÁŘ, B. *Mikrokontrolerový modul pro vývojový kit MLC interface s TMS320F2812*, 2013.
- [A40] KOŠAN, T., MOLNÁR, J., STREIT, L. *Čtyřúrovňový napěťový střídač s plovoucími kondenzátory (4L-FLC)*, 2013.
- [A41] MOLNÁR, J., KOŠAN, T., STREIT, L. *Čtyřúrovňový napěťový usměrňovač s plovoucími kondenzátory (4L-FLC)*, 2013.
- [A42] MOLNÁR, J., KOŠAN, T., STREIT, L. *Čtyřúrovňový nepřímý frekvenční měnič s plovoucími kondenzátory (4L-FLC)*, 2013.
- [A43] MOLNÁR, J., KOŠAN, T., STREIT, L. *Sestava tříhladinového NPC frekvenčního měniče*, 2013.

### **Seznam autorových výzkumných zpráv (chronologicky)**

- [A44] KOŠAN, T. *Nízkoúrovňový kontrolér pro čtyřúrovňový měnič implementovaný v CPLD*. Plzeň : Západočeská univerzita v Plzni, FEL, 2011. Číslo zprávy: 22190-015-2011.
- [A45] KOŠAN, T. *JTAGv5 - JTAG emulátor nové generace*. Plzeň : Západočeská univerzita v Plzni, FEL, 2012. Číslo zprávy: 22160-002-2012.

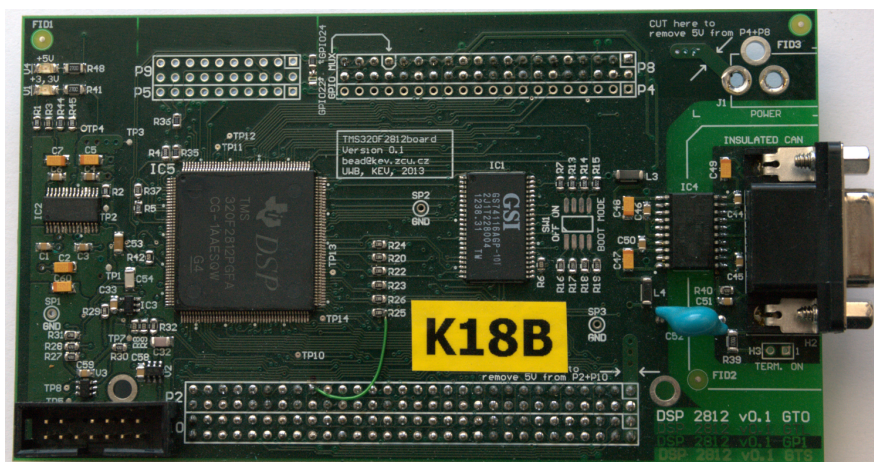
- [A46] KOŠAN, T. *MLC interface - vývojový kit pro víceúrovňové měniče s procesorem a FPGA*. Plzeň : Západočeská univerzita v Plzni, FEL, 2012. Číslo zprávy: 22190-010-2012.
- [A47] KOŠAN, T. *Nízkoúrovňový kontrolér pro tříúrovňový NPC/ANPC měnič implementovaný v CPLD*. Plzeň : Západočeská univerzita, Plzeň, 2012. Číslo zprávy: 22190-033-2012.
- [A48] KOŠAN, T. *Střídač 4L-FLC s aktivním balancováním: Odladění pro vyšší napětí a proudy*. Plzeň : Západočeská univerzita v Plzni, FEL, 2012. Číslo zprávy: 22190-046-2012.
- [A49] ZEMAN, M., JANÍK, D., KOŠAN, T., PEROUTKA, Z. *Aktivní balancování 4L-FLC měniče s využitím samostatné balancovací jednotky*. Plzeň : Západočeská univerzita v Plzni, FEL, 2012. Číslo zprávy: 22190-071-2012.
- [A50] JANÍK, D., KAMENICKÝ, P., KOŠAN, T., PEROUTKA, Z. *Studie metod přednabíjení plovoucích kondenzátorů FLC měničů*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22190-002-2013.
- [A51] JANÍK, D., KOŠAN, T., PEROUTKA, Z. *Test vybrané metody přednabíjení plovoucích kondenzátorů na prototypu FLC měniče*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22190-016-2013.
- [A52] KAMENICKÝ, P., JANÍK, D., KOŠAN, T., PEROUTKA, Z. *Vektorové řízení asynchronního motoru s měničem 4L-FLC*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22190-055-2013.
- [A53] KOŠAN, T. *Mikrokontrolérový modul TMS320F28335 pro MLC interface*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22160-002-2013.
- [A54] KOŠAN, T. *Mikrokontrolérový modul TMS570LS3137 pro MLC interface*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22160-003-2013.
- [A55] KOŠAN, T. *Univerzální měřicí desky pro víceúrovňové měniče*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22190-042-2013.
- [A56] KOŠAN, T., BEDNÁŘ, B. *Mikrokontrolérový modul TMS320F2812 pro MLC interface*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22160-006-2013.
- [A57] PEROUTKA, Z., JANÍK, D., KOŠAN, T. *Optimalizovaný návrh PD-PWM modulátoru pro MLC interface*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22190-001-2013.
- [A58] UZEL, D., BLAHNÍK, V., KAMENICKÝ, P., KOŠAN, T. *Pohon s asynchronním motorem: Experimentální studie se čtyřúrovňovými měniči*. Plzeň : Západočeská univerzita v Plzni, FEL, 2013. Číslo zprávy: 22190-037-2013.

## Seznam autorova softwaru (chronologicky)

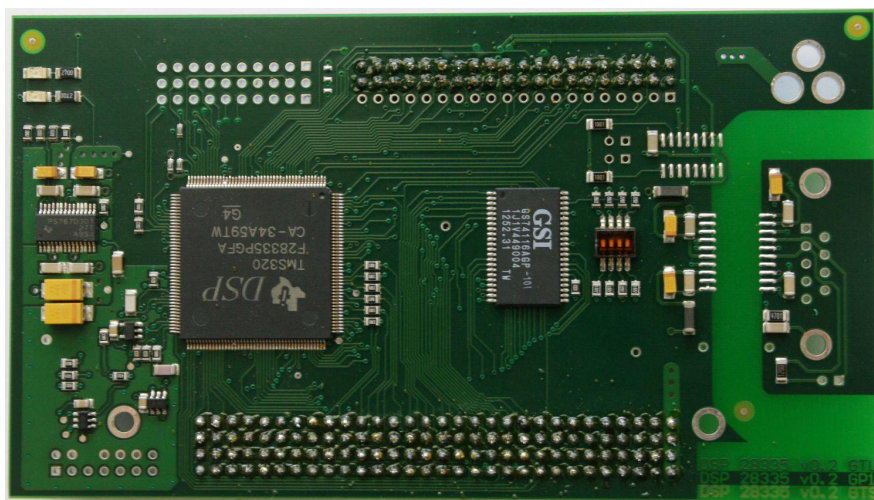
- [A59] KOŠAN, T. *PLD loader*. 2012.
- [A60] KOŠAN, T. *RALM - RICE Altium Library Manager*. 2012.
- [A61] KOŠAN, T. *RICE stock*. 2012.
- [A62] KOŠAN, T., CHLÁDEK, S. *RATS DB - RICE Automated Test System database frontend*. 2012.
- [A63] JANÍK, D., KOŠAN, T., PEROUTKA, Z. *Univerzální blokový modulátor pro řízení víceúrovňových měničů implementovaný v obvodu FPGA*. 2013.



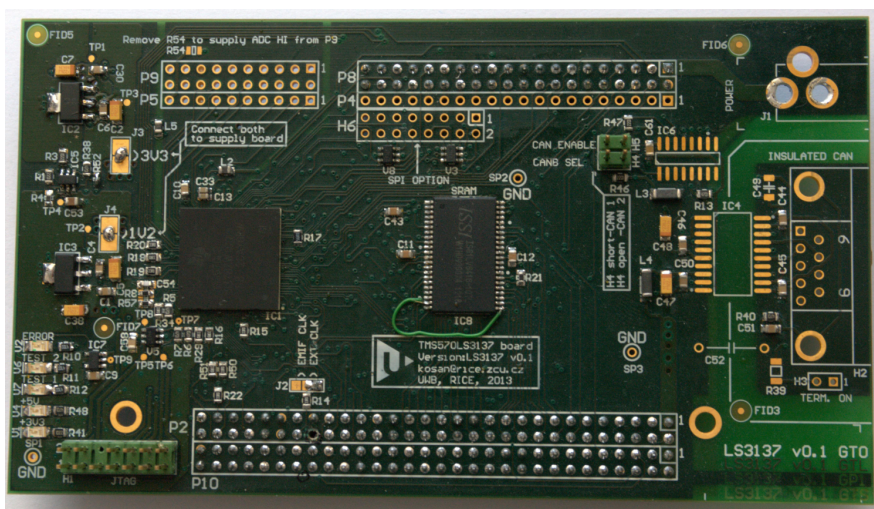




Obr. A.2.: Mikrokontrolerový modul TMS320F2812

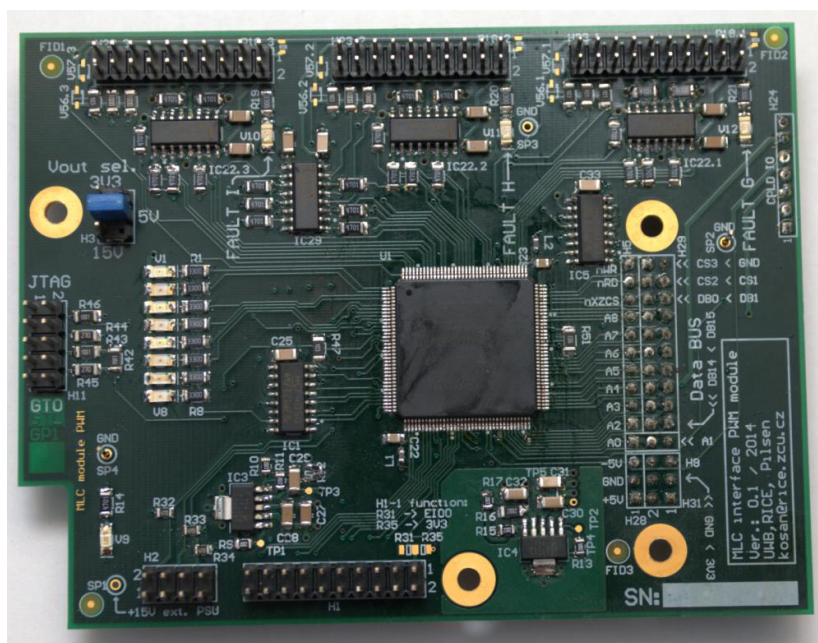


Obr. A.3.: Mikrokontrolerový modul TMS320F28335

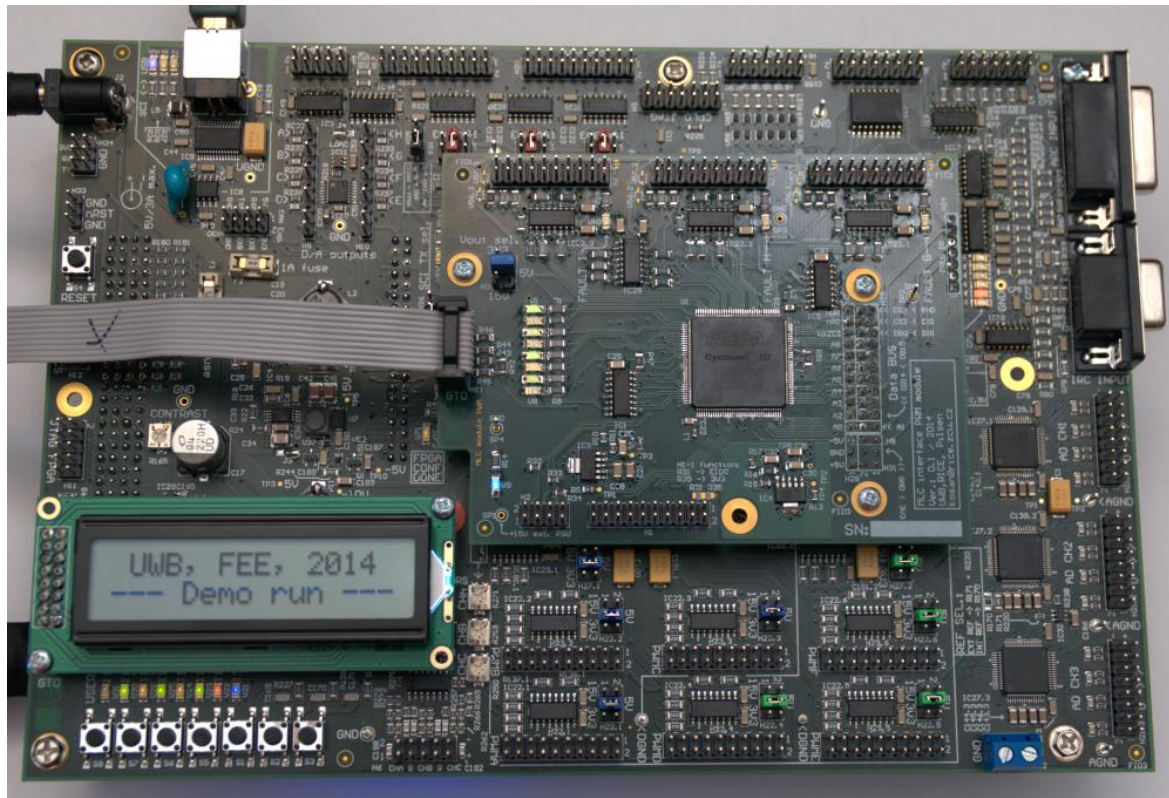


Obr. A.4.: Mikrokontrolerový modul TMS570LS3137

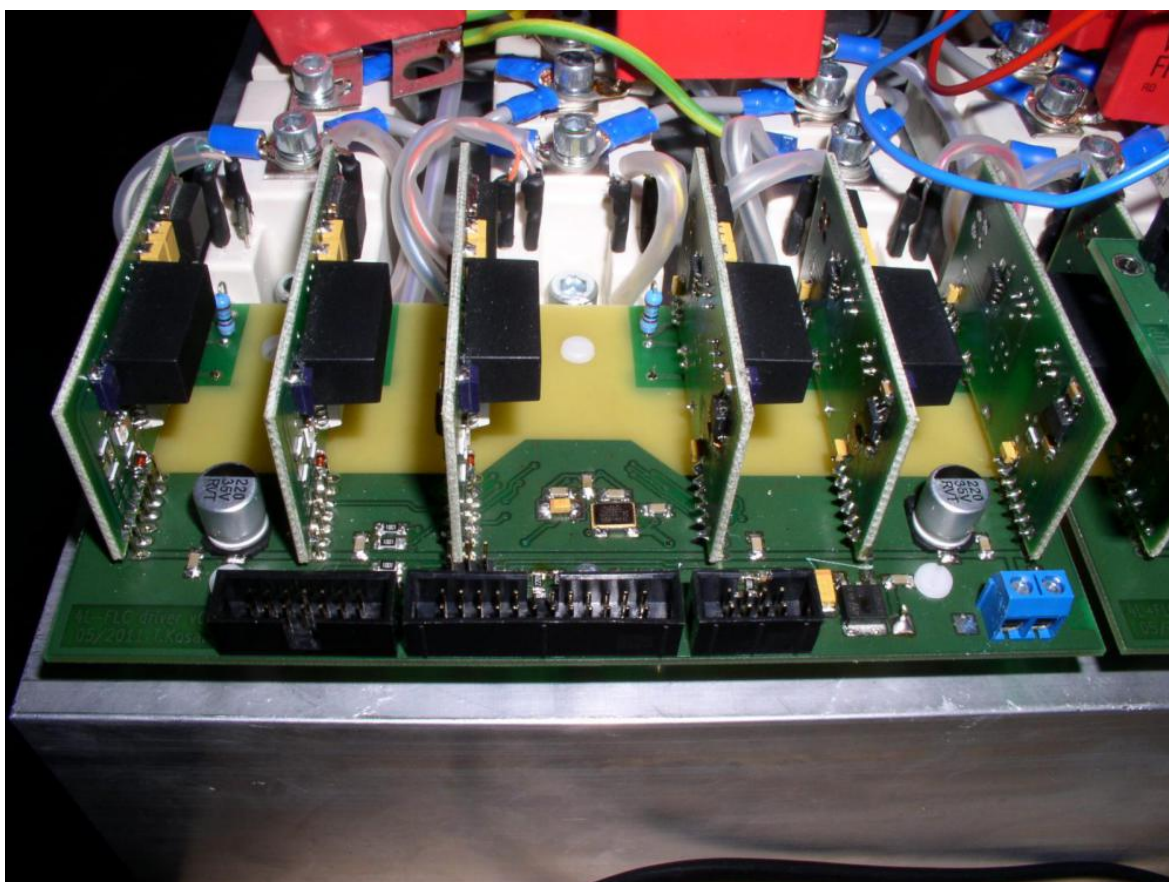




Obr. A.5.: PWM modul pro MLC interface



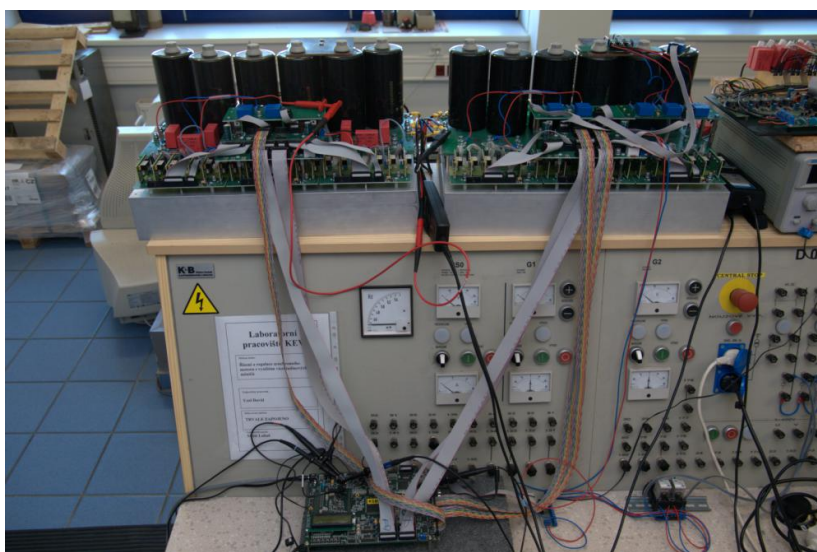
Obr. A.6.: Rozšiřující PWM modul osazený na MLC interface



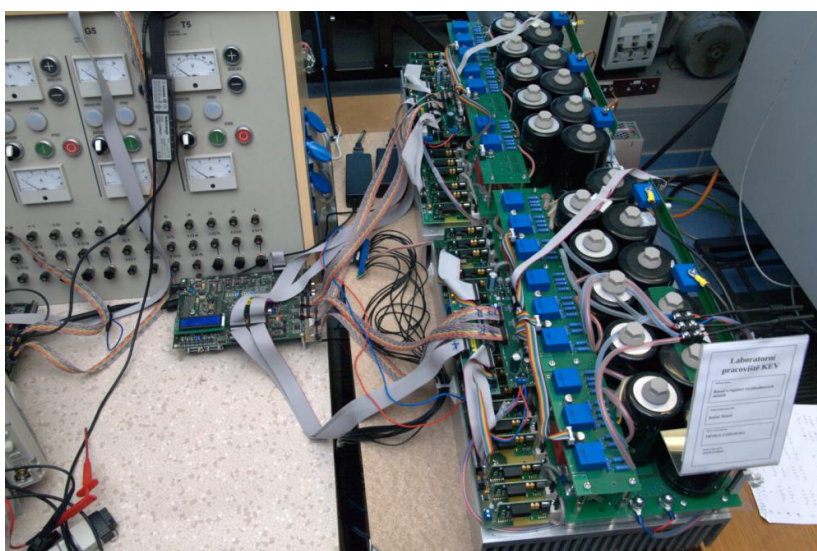
Obr. A.7.: Modul pro víceúrovňové měniče s CPLD (to je na spodní straně PCB)



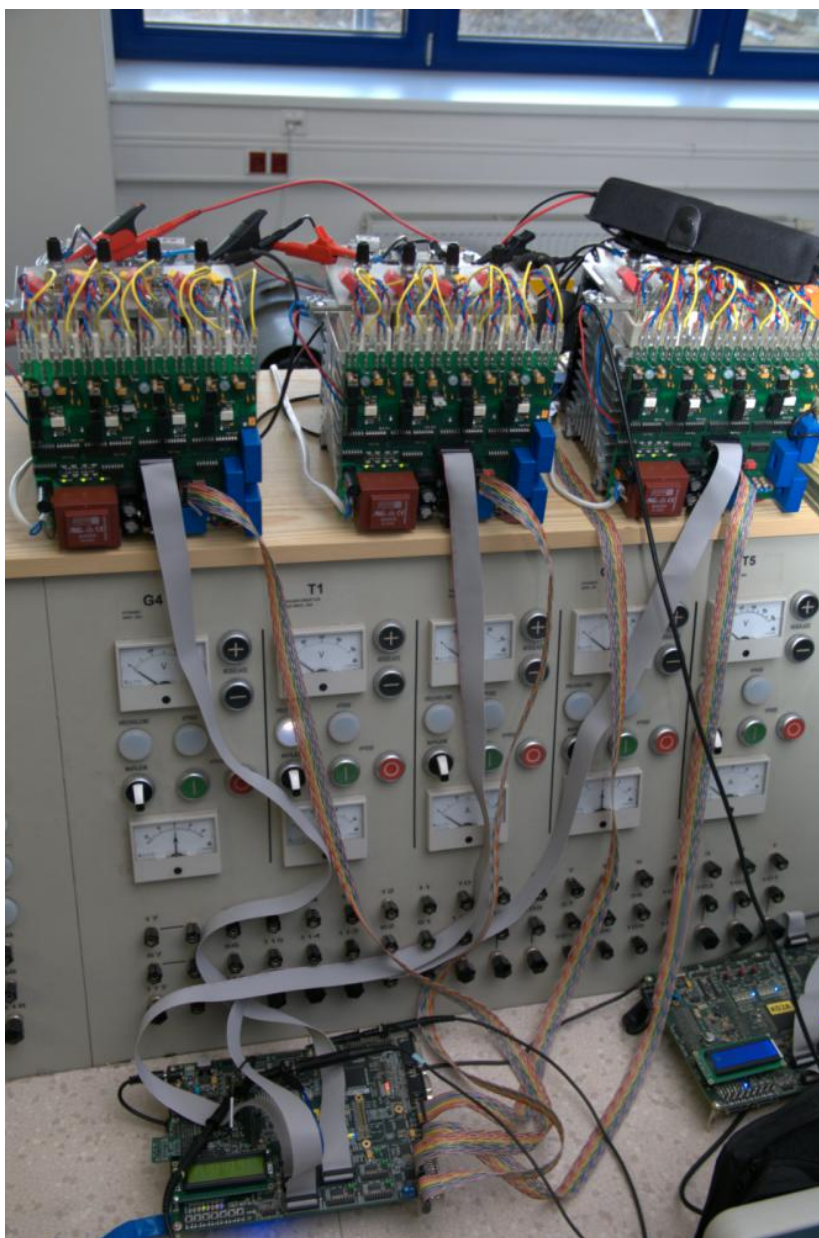
## Příloha B - Měřicí pracoviště s víceúrovňovými měniči



Obr. B.1.: Číslcový regulátor MLC interface připojený k 3L-NPC



Obr. B.2.: Číslcový regulátor MLC interface připojený k 4L-FLC



Obr. B.3.: Číslicový regulátor MLC interface připojený k 7L-CHB

## Příloha C - Parametry použitých motorů

Tab. C.1.: Parametry synchronního motoru

Jmenovitý výkon	$P_N = 10 \text{ kW}$
Jmenovité otáčky	$n_N = 1500 \text{ min}^{-1}$
Počet pólů	$p_p = 2$
Jmenovitý efektivní proud motoru	$I_{sN} = 18 \text{ A}$
Jmenovitý proud buzení	$I_{rN} = 2 \text{ A}$
Odpor statorového vinutí	$R_s = 1,15 \Omega$
Odpor rotorového vinutí	$R_r = 0,28 \Omega$
Statorová indukčnost v ose $d$	$L_{sd} = 0,09 \text{ H}$
Statorová indukčnost v ose $q$	$L_{sq} = 0,025 \text{ H}$

Tab. C.2.: Parametry asynchronního motoru

Jmenovitý výkon	$P_N = 11 \text{ kW}$
Jmenovité otáčky	$n_N = 1460 \text{ min}^{-1}$
Počet pólů	$p_p = 2$
Jmenovitá statorová frekvence	$f_{sN} = 50 \text{ Hz}$
Účinník	$\cos \varphi = 0,84$
Jmenovitý efektivní proud motoru	$I_{sNef} = 12,4 \text{ A}$
Odpor statorového vinutí	$R_s = 1,15 \Omega$
Odpor rotorového vinutí	$R_r = 1,456334 \Omega$
Hlavní indukčnost	$L_h = 0,219073 \text{ H}$
Rozptylová indukčnost statoru	$L_{ss} = 0,008921 \text{ H}$
Rozptylová indukčnost rotoru	$L_{rs} = 0,011297 \text{ H}$