

Електротехнички факултет универзитета у Београду



Дипломски рад

**Пројектовање прилагодних електронских
кола за галванску изолацију сигнала у
електромоторном погону**

Предмет: Микропроцесорско управљање електромоторним
погонима

Професор: др Слободан Вукосавић

Студент: Обрад Ђорђевић

Асистент: мр Жељко Пантић

Број индекса: 4/02

Београд

Јул, 2008

САДРЖАЈ

1. УВОД	3
2. ТЕОРИЈСКИ ОСНОВИ.....	6
3. ПРИЛАГОДНО-КОМУНИКАЦИОНА КАРТИЦА	9
3.1. Паралелни (LPT) порт РС рачунара.....	10
3.2. Интерфејс прилагодено-комуникационе картице	13
3.2.1. Конектор JDP1	13
3.2.2. Конектор J1	14
3.2.3. Конектор J2	15
3.2.4. Конектор JP1	16
3.2.5. Конектор JDR2	16
3.2.6. Конектор JDR1	17
3.3. Коришћене компоненте	19
3.3.1. Xilinx® Spartan XCS20-VQ100C	19
Структура FPGA чипа	19
CLB (Configurable Logic Block) блокови.....	20
Улазно/излазни блокови – IOB (<i>Input/Output Blocks</i>).....	20
Рутирајући канали	21
Додатни – специфични блокови FPGA чипа	21
Осцилатор на чипу.....	21
Boundary Scan.....	21
Readback	22
Startup.....	22
Конфигурација FPGA чипа.....	22
Опис пинова	23
3.3.2. Xilinx-ов PROM XC18V00	23
3.3.3. MAX525	24
3.3.4. MAX1202	25
3.3.5. SN65LBC173	26
3.3.6. LM317	27
4. ПРИЛАГОДНО-ИЗОЛАЦИОНА КАРТИЦА	28
4.1. Изолација аналогних сигнала од рачунара ка мотору	29
4.1.1. Изолација аналогних сигнала коришћењем линеарног оптокаплера IL300	31
4.1.2. Изолација аналогних сигнала коришћењем изолационог појачавача HCPL-7840	34
4.2. Изолација аналогних сигнала од мотора ка рачунару	36
4.3. Изолација дигиталних сигнала од рачунара ка мотору	40
4.4. Изолација дигиталних сигнала од мотора ка рачунару	43
4.5. Спецификације конектора прилагодено-изолационе картице	45
4.5.1. J2 конектор	45
4.5.2. J3 конектор	46
4.5.3. J1 конектор	46
4.6. <i>Layout</i> реализоване прилагодено-изолационе картице	47
5. Начин коришћења „Нове Вектре“	49
5.1. Реализовани софтвер и начин његовог коришћења	49
5.2. Процедура укључења и искључења ”Нове Вектре”, проблеми који се јављају при томе и решења тих проблема	52
5.3. Повезивање система у конфигурацију ”Нова Вектра” и враћање у конфигурацију ”Стара Вектра”	54
6. ЗАКЉУЧАК	55
7. Прилози.....	56
7.1. ПРИЛОГ А - Паралелни (LPT) порт РС рачунара.....	56

7.2. ПРИЛОГ В – Функција имплементирана на FPGA чипу	60
7.2.1. Увод.....	60
7.2.2. Top-level schematic file	60
7.2.3. LPTCOM	62
7.2.4. LCDCON.....	63
7.2.5. DEBOUNCE	68
7.2.6. DACCIP2.....	68
7.2.7. DAC2.....	69
7.2.8. DACSPI	71
7.2.9. ADCCIP.....	72
7.2.10. ADC.....	72
7.2.11. ADCSPI	74
7.2.12. ENK_CEO	75
7.2.13. KASNJENJE	76
7.2.14. ENKODER	77
7.2.15. Временски дијаграми	78
7.3. ПРИЛОГ С – Списак компоненти прилагодно-изолационе картице	80
7.4. ПРИЛОГ D – Упознавање са софтвером	81
8. ЛИТЕРАТУРА	97

1. УВОД

Галванска изолација је неопходна у свим електронским системима у којима се жели постићи висок степен сигурности и заштите једног дела уређаја од појаве пренапона у другом делу. Она је практично неопходна у уређајима са више различитих референтних нивоа за напоне тј. у уређајима са више различитих нивоа маса. Рад се бави реализацијом галванских изолација, како за аналогне, тако и за дигиталне сигнале, у једном типичном систему са више различитих нивоа маса, у коме је галванска изолација неопходна. Тај систем је рачунаром управљани електромоторни погон са асинхроним мотором.

Пројектовање и реализација галванске изолације чини први и основни део овога рада, али он рад садржи и други, додатни, део који се бави софтвером за покретање поменутог погона. Реализовани софтвер је једноставан, конзолног је интерфејса, и служи како подршка за тестирање реализованог хардвера. Уз то реализовани софтвер имплементира и брзински регулатор.

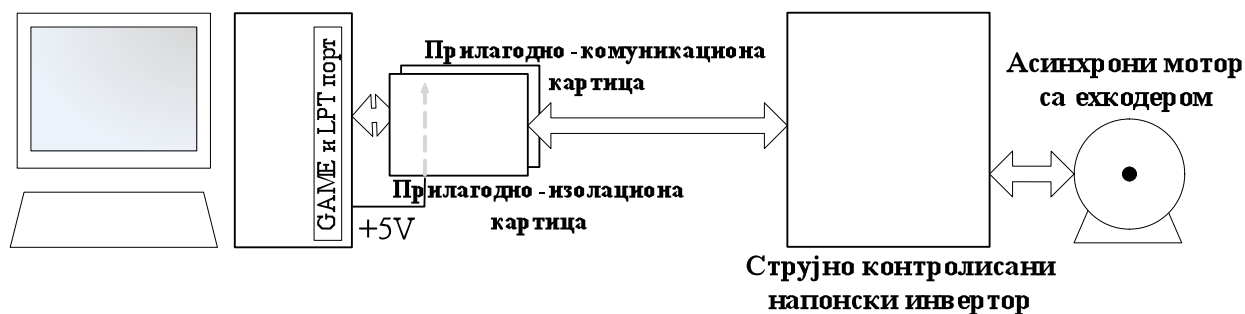
Систем који је испитиван и за који је превасходно и пројектована галванска изолација је постројење лабораторијске станице „Вектра“ која се налази у лабораторији за Микропроцесорско управљање електромоторним погонима, на Електротехничком факултету Универзитета у Београду. Наравно, испројектована картица за галванску изолацију се може користити и у другим уређајима сличне структуре. Поред галванске изолације испројектована картица врши и одговарајуће прилагођење напонских нивоа сигнала, тако да је у тексту коришћени назив за ново реализовану картицу – прилагодено-изолациона картица.

Испројектована прилагодено-изолациона картица је један у низу пројеката за осавремњивање индиректно векторски управљаног погона – лабораторијске станице ”Вектра”. Свеукупни циљ пројектовања прилагодено-изолационе картице и њеног припајања са прилагодено-комуникационом картицом са FPGA чипом, је да се рачунар старије генерације, који тренутно служи за управљање мотором, замени са савременим РС рачунаром, који би помоћу свог паралелног LPT порта (*Line Printer Terminal*) задавао команде и добијао повратне сигнале из система и на тај начин управљао радом асинхроног мотора. Затечено стање лабораторијске станице детаљно је описано у упутству за вежбу „Лабораторијска станица за испитивање алгоритма управљања индиректно векторски управљаним асинхроним мотором – Вектра“ из 2001. године, те овде неће бити додатно описивано.

Поменути прилагодно-комуникациона картица је управо, један од претходних пројеката лабораторије за Микропроцесорско управљање електромоторним погонима. У овом раду је искоришћена универзалност и широка функционалност ове картице, коју јој пре свега даје коришћени FPGA чип и AD и DA конвертори, као и комуникација према рачунару путем стандардног паралелног порта.

Овај рад је имао за циљ да сигнале, потребне за управљање електромоторним погоном, постојеће прилагодно-комуникационе картице, галвански изољује, и изврши потребно прилагођење напонских нивоа тих сигнала. Како реализована картица само допуњава ”недостатке” претходно постојеће прилагодно-комуникационе картице, то се дошло на идеју да реализована прилагодно-изолациона картица буде физички припојена уз прилагодно-комуникациону картицу. На тај начин је практично добијена једна универзална ”сендвич” картица (која се заправо састоји из две картице), која се може директно повезати на паралелни порт рачунара, а која може преносити у оба смера и аналогне и дигиталне сигнале и при том их још и галвански изоловати. На тај начин ова картица омогућава и неке друге видове управљања мотором, поред тренутно постојећег управљања задавањем струјних референци са хистерезисном регулацијом, као што је управљање помоћу импулсно-ширинске модулације (PWM).

Дакле рад се бави повезивањем одређеног модула тј. прилагодно-комуникационе картице са рачунаром, њиховом комуникацијом и повезивањем са друге стране са одговарајућим контролним улазима односно мереним излазима мотора као система, преко новореализоване прилагодно-изолационе картице. Комуникација према рачунару је остварена преко стандардног паралелног LPT порта. Блок шема целокупног система приказана је на следећој слици (Слика 1.1).



Слика 1.1 Блок шема система

Приликом реализације интерфејса ка мотору, нове картице, водило се рачуна о томе да систем може без пуно преправки, једноставним превезивањем конектора, радити и са старим картицама.

Иначе, циљ овога рада није био само реализација поменутог хардвера и софтвера, већ и прављење што боље документације која би могла послужити у реализацијама нових идеја за које нови хардвер оставља доста простора. Како сама концепција рада не би била поремећена, већина тих додатних материјала је смештена у прилоге на крају рада. Овај дипломски рад, треба да послужи као приручник свакоме ко буде желео да настави на

усавршавању, преправкама и поправкама како хардвера, тако и софтвера лабораторијске станице „Вектра“.

Рад читаоца, на почетку упознаје са теоријским основима индиректног векторског управљања. Ово поглавље само у најкраћим цртама излаже тематику овог проблема, како би се могла стећи представа који су то управљачки сигнали које рачунар шаље ка систему, а који су то мерни сигнали – повратне информације, које се враћају ка рачунару.

Потом следи поглавље за упознавање са хардвером прилагодено-комуникационе картице. Ту је описан LPT порт, као интерфејс преко кога се врши комуникација између рачунара и прилагодено-комуникационе картице, затим је описан интерфејс саме картице (спецификације конектора), потом су у кратким цртама описани сви чипови који се налазе на картици и на крају је дат кратак опис функције која је имплементирана на FPGA чипу, као централној јединици прилагодено-комуникационе картице.

Изолациони део или прилагодено-изолациона картица, која се физички налази „приљубљена“ уз прилагодено-комуникациону картицу, описана је у наредном поглављу, и она заправо представља централни део рада, односно то је ново направљена картица додата у систем. Циљ, ове картице је да изврши усклађивање напонских нивоа са стране рачунара, тј. прилагодено-комуникационе картице, према нивоима сигнала на постојећем постројењу које омогућава покретање мотора. Поред прилагођења реализована картица обезбеђује и неопходну галванску изолацију свих сигнала који се преносе. Уз сваку од реализација, наведене су и карактеристике добијене након мерења.

Следеће поглавље представља опис реализованог софтвера, његове могућности, начин задавања команди и начин укључивања лабораторијске станице „Вектра“. У овом поглављу дата су и упутства како се једноставним превезивањем каблова, систем може вратити у стање у којем је био пре израде овог дипломског рада, односно у стање описано у упутству за лабораторијску вежбу „Лабораторијска станица за испитивање алгорита управљања индиректно векторски управљаним асинхроним мотором – Вектра“.

На крају је дат закључак о квалитету реализованог хардвера и могућностима порширења и имплементацији нових функција на постојећем постројењу.

Прилози представљају посебан део овога рада, с обзиром да могу бити јако корисни, свакоме ко буде желео да још нешто надогради у постојећем систему.

- Прилог А детаљно описује паралелни – LPT порт рачунара.
- Прилог В детаљно објашњава функцију имплементирану на FPGA чипу, и даје потребна упутства за његово репрограмирање.
- Прилог С даје листинг компоненти које се налазе на прилагодено-изолационој картици.
- Прилог D представља искоментарисани и додатно појашњени .C код, који служи за тестирање реализованог хардвера.

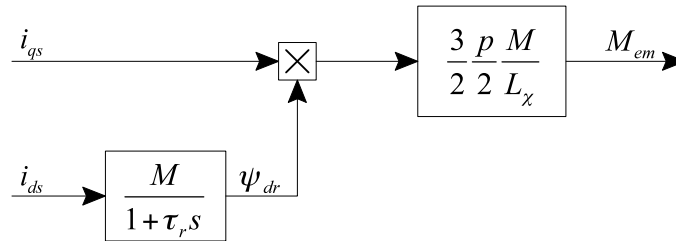
2. ТЕОРИЈСКИ ОСНОВИ

Значај асинхроних машина је велики, а примена разноврсна. О њиховој заступљености и значају најбоље говори податак да асинхроне машине троше до 50% укупно произведене електричне енергије [1]. Асинхроне машине, међутим, имају један велики недостатак, а то је да мрежом напајана асинхрона машина има тачно дефинисану брзину обртања, те се као таква не може користити у апликацијама где се захтева регулација брзине и велика прецизност покретачког момента. Једно од решења тог проблема даје концепт индиректног векторског управљања, који је имплементиран на лабораторијској станици ”Вектра”.

Лабораторијска станица ”Вектра” демонстрира индиректно векторско управљање асинхроним мотором. Овај концепт по угледу на машине једносмерне струје, где се моментом директно може управљати задавањем струје арматуре, покушава да и од асинхроне машине начини линеарни извршни орган са независном побудом. Идеја је да се трофазна асинхрона машина напаја из струјно регулисаног напонског инвертора, који ће генерисати одговарајуће напоне u_a , u_b и u_c , на статору машине, који ће произвести струју статора тачно одговарајуће амплитуде, учестаности и фазног става.

Из теорије знамо да се свака трофазна машина може приказати двофазним еквивалентом употребом Кларкине трансформације. Такође потом се у теорији при разради динамичког модела асинхроне машине, уводи синхрони ротирајући (dq) координатни систем – Паркова трансформација, и на тај начин се превазилази проблем нестационарне матрице индуктивности, тј. нестационарног модела, који је у претходном моделу постојао и у стационарном стању. Из теорије се такође показује да, уколико једну осу синхроног ротирајућег система (нпр. d -осу) поставимо у правцу флуksа ротора (односно, уколико постигнемо да је пројекција флуksа ротора на другу (q -осу) осу једнака нули), да ћемо добити модел асинхроне машине, у коме је као и у моделу једносмерне машине, распрегнуто управљање моментом и флуksом. Према добијеном моделу, флуks ротора асинхроне машине $\Psi_r = \Psi_{dr}$, директно је пропорционалан пројекцији струје статора на d -осу синхроног ротирајућег координатног система - i_{ds} . Такође, добија се да је вредност момента асинхроне машине пропорционална производу флуksа ротора и q компоненте струје статора - i_{qs} . Како су промене флуksа јако споре, а и по угледу на машине једносмерне струје, често се вредност флуksа ротора одржава константном, па се у том случају може рећи да пројекција статорске струје на d -осу управља вредношћу флуksа ротора, а да пројекција статорске струје на q -осу управља вредношћу момента асинхроне машине. Модел асинхроног мотора у коме је

остварено распрегнуто управљање моментом и флуksom приказан је на следећој слици (Слика 2.1) [2].



Слика 2.1 Динамички модел асинхроног мотора са распрегнутим управљањем моментом и флуksom

Можемо закључити да индиректно векторско управљање превазилази проблем уске примене асинхроних машина, због немогућности промене њене брзине ротације уколико се директно напаја из мреже, већ се векторски управљана асинхрона машина понаша као линеарни извршни орган погодан и у апликацијама са строгим захтевима за регулацијом брзине и управљачког момента, попут серво апликација и електричне вуче.

Да бисмо у пракси дошли до оваквог модела, потребно је да обезбедимо хардвер или софтвер на РС-ју, који врши Кларкину и Паркову трансформацију, да обезбедимо драјвере за паљење транзистора у струјно контролисаном напонском инвертору, да обезбедимо струјне сензоре, који ће мерити струје статора, и коначно да обезбедимо уређај који ће нам давати информацију о позицији флуksа ротора. Метода индиректног векторског управљања, позицију роторског флуksа одређује, како јој и само име каже, индиректним методама, и то као збир измерене позиције ротора, и у рачунару у напред процењеног клизања. Дакле видимо да је за методу индиректног векторског управљања, у пракси неопходан давач позиције, а у конкретној реализацији се као давач користи енкодер.

Регулација управљања и инверзна Паркова трансформација се обављају у софтверу РС рачунара, док се Кларкина, тј. инверзна Кларкина трансформација обавља хардверски, аналогним сабирачима. Као струјни сензори тренутно се користе шантови, а планира се увођење LEM сензора, ради додатне галванске изолације.

Коначно можемо закључити да скуп каблова, односно информација које се размењују између рачунара и струјно регулисаног напонског инвертора чине:

- *референтне вредности задатих* струја које рачунар након обраде података у dq координатном систему инверзном Парковом трансформацијом преводи у $\alpha\beta$ координатни систем и те референтне вредности за струје i_α^* и i_β^* прослеђује ка струјно регулисаном напонском инвертору.

- *фаза са енкодера*. Рачунар са давача позиције добија информацију о угаоној позицији ротора мотора. У конкретној реализацији налази се енкодер који шаље две фазно померене поворке импулса (фаза А и фаза В).

Ово представља првобитни скуп сигнала који се преносе, односно скуп сигнала који је потребан за индиректно векторско управљање са хистерезисном регулацијом, која је првобитно била једини могући вид регулације. Како прилагодено-комуникациона картица даје читав низ могућности за различите врсте управљања и разноразно прикупљање података, то се у новој реализацији поред поменутих података преноси још:

- 6 дигиталних сигнала од рачунара ка струјно регулисаном напонском инвертору, чија је првобитна намена за PWM (*Pulse Width Modulation*) управљање мотором

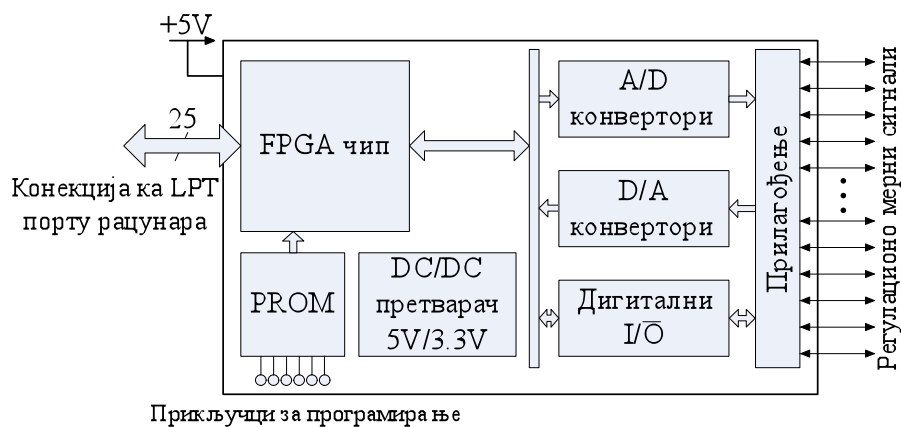
- 4 аналогна сигнала од инвертора ка рачунару, чија је примарна намена за пренос података о измереним струјама у фазама мотора.

Додатни, поменути сигнали, потребно је да поред постојећег управљања са хистерезисном регулацијом обезбеде услове и за реализацију управљања PWM-ом, али су додати сигнали опште намене па се могу користити и у неке друге сврхе.

3. ПРИЛАГОДНО-КОМУНИКАЦИОНА КАРТИЦА

За било каква разматрања, нове реализоване прилагодено-изолационе картице, и принципа рада целокупног система, па чак и за разматрање коришћеног покретачког софтвера, неопходно је детаљно упознавање са хардвером прилагодено-комуникационе картице, која представља спрегу између рачунара и мотора.

Блок шема прилагодено-комуникационе картице приказана је на следећој слици (Слика 3.1).



Слика 3.1 Блок шема прилагодено-комуникационе картице

У овом поглављу су детаљно описани:

LPT порт, као интерфејс преко кога се врши комуникација између рачунара и прилагодено-комуникационе картице; затим

Интерфејс прилагодено-комуникационе картице (спецификације конектора),

Коришћене компоненте. У кратким цртама описани су сви чипови који се налазе на картици.

Функција имплементирана на FPGA чипу као централној јединици прилагодено-комуникационе картице.

За наредно поглавље издвојена је прилагодено-изолациона картица, као централни део овог дипломског рада, док хардвер самог струјно контролисаног напонског инвертора неће бити додатно описиван с обзиром да се детаљна објашњења и електричне шеме налазе у упутству за лабораторијску вежбу [1].

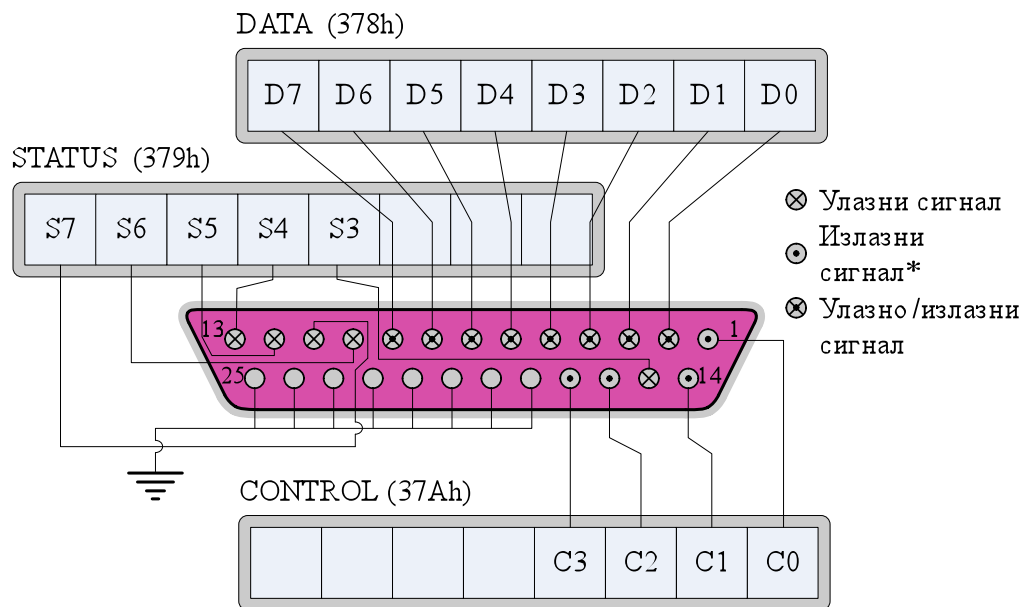
3.1. Паралелни (LPT) порт РС рачунара

Паралелни интерфејс – LPT порт, представља стандардни интерфејс већине савремених РС рачунара, па је с тога и изабран при реализацији, како би заменио тренутно постојећу специфичну картицу која је повезана на ISA магистралу, која се све ређе среће код савремених рачунара.

По оригиналној концепцији и реализацији пренос података преко овог интерфејса био је могућ само у једном смеру од рачунара ка периферији – штампачу, за који је првобитно направљен овај интерфејс. Такође пренос података путем овог порта у почетку није специфицирао никакве стандарде. Данас, срећом, то није тако, па је пренос могућ у оба смера, а асоцијација IEEE је донела стандард IEEE-P1284 којим је специфицирано 5 могућих начина рада овог интерфејса. Нешто више о историјату, спецификације преноса и детаљна шема LPT порта, дати су у 7.1, на крају овога рада.

Без обзира на постојеће стандарде, пренос података не мора бити ни према једном од дефинисаних модова рада, већ се целокупни порт може посматрати само као скуп регистара у које се нешто уписује, или из којих се нешто чита. Управо један такав начин преноса коришћен је у самом раду.

Програмирање паралелног порта се врши уз помоћ три 8-битна регистра. Складно са линијама паралелног порта, ти регистри су: регистар података – *DATA Register*, статусни регистар – *STATUS Register* и управљачки или контролни регистар – *CONTROL Register*, и налазе се на три суседне адресе у улазно/излазном (*Input/Output* – $I/\bar{O}$) $I/\bar{O}$ адресном простору. Конкретно, за LPT1 порт $I/\bar{O}$ адресе тих регистара су од 378h до 37Ah респективно. Улази односно излази ових регистара су на одговарајући начин повезани са одговарајућим пиновима самога порта, као што је то приказано на следећој слици (Слика 3.2).



Слика 3.2 Паралелни (LPT) порт рачунара и његови регистри

Пинови који су на слици обележени као излазни, су заправо код новијих генерација улазно/излазни, али овде ће се користити само као излазни, па су зато на слици тако и обележени. Називи свих пинова (бита регистара) су смишљено названи општим именима (C0-C3, S3-S7, D0-D7). То је учињено са намером да се нагласи да су заправо сви пинови опште намене и да њихови називи не би асоцирали на везу са штампачем као периферијом. У следећој табели (Табела 3.1) приказана је функција додељена, при реализацији, свакој од линија паралелног порта.

Табела 3.1 Додељене функције линија паралелног интерфејса РС рачунара

Група	Општи назив	Смер	Функција у реализацији
Линије података	D0-D7		Бајтни пренос података за D/A конверуију.
Управљачке линије	C0	Од рачунара ка картици	Сигнали C0 и C1отпочињу и врше пренос података у оба смера. Они стартују све процесе на периферији, и врше њихову синхронизацију. Основна улога C1 је тактовање, а C0 отпочињање преноса (видети детаљније у поглављу 7.2)
	C1		
	C2		
	C3		
	S6	Од картице ка рачунару	Не користи се
Статусне линије	S7		Прослеђује се на излазни пин 86 FPGA чипа и на њему се може мерити време трајања прекида
	S5		
	S4		
	S3		

Сваки, стандардом дефинисани мод преноса, предвиђа одређену синхронизацију преноса и проверу да ли су подаци коректно пренети, док у методу коришћеном у раду нема никакве сигнализације и провере. Исправност преноса података заснива се на добро процењеном тајмингу интерфејса самога порта и саме периферије. Односно уколико, на пример, шаљемо податке од рачунара ка периферији, пре слања наредног податка нећемо вршити проверу да ли је претходни прихваћен, већ ћемо тај наредни податак послати након одговарајућег времена за које смо сигурни да је периферија успела да прихвати и обради претходни податак. Такође и сигурност преноса података од периферије ка рачунару се врши на исти начин, односно без додатне синхронизације већ само на основу процене брзине рада самих уређаја. Такав вид комуникације је намерно изабран, са циљем да пренос података буде што бржи.

Комуникација између прилагодено-комуникационе картице и рачунара врши се методом прекида. Имплементирани FPGA чип на прилагодено-комуникационој картици, шаље захтев за прекид рачунару путем линије $\overline{\text{ACK}}$ (S6) на сваких 220 μs (заправо ово време је програмабилно и задаје се у самој коду, а 220 μs је минимална вредност за коју пренос свих података исправно ради). Прекидна линија IRQ7 (*Interrupt Request*) (тип прекида – 0fh) на основној плочи рачунара намењена је за паралелни порт LPT1. Ако је дозвољен прекид IRQ7 (бит 4 управљачког – CONTROL регистра 37Ah порта, има вредност логичко 1 и бит 7 у регистру IMR (*Interrupt Mask Register*) за маскирање прекида који се налази у контролеру прекида 8259A, има вредност 0) сигнал на линији $\overline{\text{ACK}}$ (S6) активираће захтев за прекид IRQ7. Између $\overline{\text{ACK}}$ сигнала и линије IRQ7 постоји инвертор, тако да пошто контролер прекида генерише захтев за прекид на узлазну ивицу импулса, свака силазна ивица $\overline{\text{ACK}}$ (S6) сигнала изазиваће прекид, наравно, уколико су прекиди дозвољени. С обзиром да наша периферија није штампач, већ прилагодено-комуникациона картица, део нашег задатка је да прекиду IRQ7 чији је тип 0fh у IVT (*Interrupt Vector Table*) табели прекида, и коме је додељена одређена прекидна рутина за рад штампача, заменимо са неком новом, нашом рутином (у коду – *intserv*) која ће вршити одговарајући пренос података. На тај начин, коришћењем технике прекида, и замене прекидних рутина, остварујемо комуникацију PC рачунара са нашим уређајем путем паралелног порта, и самим тим омогућујемо контролу уређаја путем рачунара, што и представља наш првобитни циљ.

Паралелни порт PC рачунара је 25-пински, а сам конектор је типа DB-25. Осам линија, излазних (односно бидирекционих), је за пренос самих података, четири излазне линије служе за контролу рада периферије, пет улазних линија враћају податке о њеном статусу, а осам линија је физички повезано на масу. Сигнали који се преносе по линијама су стандардног TTL напонског нивоа ($V_{cc} = 5V$, $V_{ol} = 0.4V$, $V_{oh} = 2.4V$, $V_{il} = 0.8V$, $V_{ih} = 2V$). Излазни напон се креће у границама од -0.5V до +5.5V. Такође, и интерфејс саме периферије мора бити са одговарајућим TTL напонским нивоима. Излазне линије имају *Slew Rate* од 0.05V/ns до 0.40V/ns, при излазној отпорности од 50 $\pm$ 5 Ω , мерено до половине V_{oh} . Преносни вод (карактеристичне импедансе од 50 Ω) обично није терминисан па је пренос могућ на растојањима до 4 метра. Да би се сачувала компатибилност са првобитним *centronics* портом, који има отворени колектор као излаз, за пренос логичке нуле потребни су *pull-up* отпорници на стани периферије (препоручују се стандардне вредности 2.2k Ω или 4.7k Ω , али може бити и било која вредност). Брзина преноса, код стандардног паралелног интерфејса, је од 100 kB/s до 200 kB/s, док рецимо код усавршеног паралелног преноса EPP брзина је чак и до 10 пута

већа (1.5 до 2 MB/s). Струјни капацитет излазних линија за податке износе до 2.6 mA, док су улазне линије у стању да прихвате струју до 24 mA. Када су у питању контролне линије, те вредности износе 0.5 mA за излазну струју и 7 mA за улазну [6] [7].

Коначно, можемо закључити да су сама архитектура и начин рада паралелног интерфејса прилично једноставни. Паралелни интерфејс се не мора користити само за слање података, за шта је првобитно био намењен, већ се може посматрати као скуп од 17 линија TTL напонских нивоа, од којих се неке могу користити као улазне, неке као излазне, а неке као бидирекционе. Ове линије се могу користити засебно, или у групама, за управљање једним или више уређаја. Дакле видимо да постоји читав низ могућих начина његовог коришћења чиме се могу остварити различите корисне примене паралелног интерфејса.

3.2. Интерфејс прилагодено-комуникационе картице

Интерфејс прилагодено-комуникационе картице, сачињавају 6 конектора. У наставку је дат опис сваког од конектора, са детаљним спецификацијама и функцијом сваког од његових пинова по на особ.

3.2.1. Конектор JDP1

JDP1 је 25-пински „мушки“ DB25 конектор. Преко њега је прилагодено-комуникациона картица повезана са LPT портом рачунара. Карактеристике одговарајућих пинова конектора дате су у следећој табели (Табела 3.2). Сви улазни и излазни сигнали овог конектора су стандардног TTL напонског нивоа.

Табела 3.2 Називи и функције линија паралелног интерфејса РС рачунара

Пин	Назив	Функција	Смер	Конекција на плочи	Стр. капац.
1	C0	Контролни бит. Има улогу при иницијализацији преноса, и отпочињању многих процеса на FPGA чипу (видети 7.2)	рач-пер	Кор. деф. I/O пин 2 FPGA чипа	0.5 mA
2-9	D0-D7	Линије за пренос података за D/A конверуију	рач-пер	Кор. деф. I/O пинови 3, 7, 8, 9, 10, 13, 14, 15 FPGA чипа	2.6 mA
10	S6	Путем ове линије картица тражи прекид на сваких 220 μ s	пер-рач	Кор. деф. I/O пин 16 FPGA чипа	12 mA
11	S7	Путем пинова 11 (MSB), 12, 13 и 15 (LSB) шаљу се подаци, у нибловима, са A/D конвертора и енкодера	пер-рач	Кор. деф. I/O пин 17 FPGA чипа	12 mA
12	S5		пер-рач	Кор. деф. I/O пин 18 FPGA чипа	12 mA
13	S4		пер-рач	Кор. деф. I/O пин 19 FPGA чипа	12 mA
14	C1	Тактује пренос података. На силазну ивицу подаци се шаљу од рачунара ка периферији, док на узлазну се шаљу од картице ка рачунару (видети 7.2)	рач-пер	Кор. деф. I/O пин 20 FPGA чипа	0.5 mA
15	S3	Путем пинова 11 (MSB), 12, 13 и 15 (LSB) шаљу се подаци, у нибловима, са A/D конвертора и енкодера	пер-рач	Кор. деф. I/O пин 21 FPGA чипа	12 mA
16	C2	Не користи се	рач-пер	Кор. деф. I/O пин 27 FPGA чипа	0.5 mA
17	C3	Показује колико траје <i>Interrupt</i>	рач-пер	Кор. деф. I/O пин 29 FPGA чипа	0.5 mA
18-25	GND	Аналогна маса	-	Аналогна маса.	-

Рач-пер – Од рачунара ка прилагодено-комуникационој картици (као периферији)

Пер-рач – Од прилагодено-комуникационе картице (периферије) ка рачунару

Кор. деф. I/O пин – Кориснички дефинисани I/O пин

Стр. капац. – струјни капацитет. Као податак је навођен податак са оне стране која покреће пренос.

3.2.2. Конектор J1

J1 је 3-пински конектор помоћу којег се дефинише мод рада FPGA чипа. Пин 1 овог конектора је повезан на напајање са рачунара ($V_{cc}=5V$), а пин 3 је повезан на масу рачунара ($GND=0V$). Постављањем „*jumper-a*“ између другог и првог, односно трећег пина, дефинише се *Slave*, односно *Master Serial* мод рада FPGA чипа, респективно. У раду је, као што ће то бити описано при опису FPGA чипа (видети поглавље 3.3.1), коришћен *Master Serial* мод рада чипа, па је „*jumper*“, након програмирања, потребно поставити између другог и трећег прикључка конектора. Спецификације овог конектора и модови рада за одговарајући положај „*jumper-a*“ дате су у следећој табели (Табела 3.3).

Табела 3.3 Спецификације пинова конектора J1

Пин	Назив	Пинови које повезује „ <i>jumper</i> “	Изабрани мод рада FPGA чипа
1	Vcc	1 – 2 (MODE = Vcc)	<i>Slave Serial</i> мод
2	MODE	2 – 3 (MODE = GND)	<i>Master Serial</i> мод
3	GND		

3.2.3. Конектор J2

J2 је 6-пински конектор преко чијег првог пина се периферија снабдева напајањем Vcc од +5V, које се доводи са првог, осмог, деветог или петнаестог пина GAME порта рачунара. Пин 2 је повезан на масу, али је маса која је заједничка за PC и прилагодно-комуникациону картицу већ доведена путем 18-25 линије LPT порта, тако да овај пин конектора може остати слободан. Пинови 3 до 6 представљају пинове за програмирање FPGA чипа. Та четири пина су заправо интерфејс за *Boundary Scan* конфигурисање FPGA чипа. То су улази TCK, TDI и TMS и излаз TDO, који представљају редом трећи, пети, шести и четврти пин J2 конектора. Помоћу њих се, серијски, може извршити конфигурисање чипа, али исто тако може се извршити и читавање тренутног стања FPGA чипа (за детаље погледати одељак *Boundary Scan* из поглавља 3.3.1 Xilinx® Spartan XCS20-VQ100C).

Проблем овог начина конфигурације је то што овако конфигурисан чип, задржава своју конфигурацију, само док је под напајањем, док у случају прекида напајања и поновног укључења, потребна је и поновна конфигурација чипа.

Спецификације овог конектора дате су у следећој табели (Табела 3.1).

Јако сличну функцију овом конектору има и конектор JP2.

Табела 3.4 Спецификације пинова конектора J2

Пин	Ознака	Назив	Функција	Конекција на плочи	Стр. капац.
1	Vcc	Напајање	Путем овог пина се доводи напајање за картицу (+5V) са GAME порта рачунара	Vcc	-
2	GND	Маса	Остаје неповезан (маса је већ доведена линијама 18-25 LPT порта)	GND	-
3	TCK	<i>Test Clock</i>	Уколико се користи <i>Boundary Scan</i> за конфигурацију чипа они представљају клок, серијски излаз, серијски улаз и пин за избор тест мода, редом.	Пин 5 FPGA чипа	12mA
4	TDO	<i>Test Data Output</i>		Пин 76 FPGA чипа	
5	TDI	<i>Test Data In</i>		Пин 4 FPGA чипа	
6	TMS	<i>Test Mode Select</i>		Пин 6 FPGA чипа	

3.2.4. Конектор JP1

JP1 је 6-пински конектор са идентичним називима и функцијама пинова као и конектор J2. Разлика је у томе што се путем овог конектора не врши директно програмирање FPGA чипа, већ се подаци смештају у PROM меморију где остају трајно запамћени, без обзира на присуство напајања, чиме се отклања претходно поменути проблем. При сваком укључивању, односно стављању прилагодено-комуникационе картице под напајање, врши се серијски препис података из PROM-а у сам FPGA чип, односно врши се његово конфигурирање. Такође, разумљиво је да се путем ових пинова не може вршити читање стања чипа. Коришћени PROM подржава *Boundary Scan* логику тако да се подаци на идентичан начин преносе како директно у FPGA чип, тако и у коришћени PROM.

Пинови 1 и 2 су такође напајање и маса, али се они доводе заједно са осталим сигнаlima са одговарајућег конектора са програматора који се прикључује на паралелни порт рачунара. Треба запазити да је напон напајања PROM-а +3.3V, а да се програмирање врши напоном од +5V са програматора. Иако је напајање PROM-а мање од напона напајања FPGA чипа њихови улази и излази су међусобно компатибилни. Такође треба приметити и то да сада немамо масу са JDP1 конектора са LPT порта, пошто је сада на LPT порт повезан програматор, па је повезивање масе са програматора неопходно.

Коришћење PROM-а за програмирање FPGA чипа очигледно има пуно предности па можемо рећи да је улога конектора JP1 за програмирање чипа, док је главна улога J2 конектора за напајање чипа, иако се као што је већ речено помоћу њега може извршити и његова конфигурација.

Спецификације конектора JP1 дате су у следећој табели (Табела 3.5).

Табела 3.5 Спецификације пинова конектора JP1

Пин	Ознака	Назив	Смер	Функција	Конекција на плочи
1	Vcc	Напајање	-	Напајање са програматора	Vcc
2	GND	Маса	-	Маса са програматора	GND
3	TCK	<i>Test Clock</i>	IN	За упис програма за конфигурацију чипа у PROM. Ови сигнали представљају клок, серијски излаз, серијски улаз и пин за избор тест мода, редом.	Пин 6 PROM-а
4	TDO	<i>Test Data Output</i>	OUT		Пин 17 PROM-а
5	TDI	<i>Test Data In</i>	IN		Пин 4 PROM-а
6	TMS	<i>Test Mode Select</i>	IN		Пин 5 PROM-а

3.2.5. Конектор JDR2

JDR2 је 25-пински DB25 конектор који служи за пријем сигнала са енкодера. Са енкодера се доводе фазе енкодера, али како су сигнали који долазе са енкодера јако зашумљени, поред фаза доводе се и њихове инвертоване вредности (према стандарду ANSI RS-485). На овај начин, много је лакше отклонити шум, и добити стварни изглед сигнала фазе, то се ради довођењем сигнала и његове инвертоване вредности на хистерезисни компаратор. Излаз оваког компаратора, истог је облика као и послати сигнал. На плочи се налазе два SN65LBC173 кола која служе за отклањање шума, која имају по 4 хистерезисна

компаратора, што значи да плоча – прилагодено-комуникациона картица, подржава пријем сигнала до са 4 енкодера. Међутим у конкретној реализацији користи се само један енкодер и галванска изолација је обезбеђена само за њега. Дакле, са овог конектора ка прилагодено-изолационој картици иду 4 линије и то са пинова 9, 10, 22 и 23.

Спецификације пинова овог конектора дате су у следећој табели (Табела 3.6).

Табела 3.6 Спецификације пинова конектора JDR2

Пин	Ознака	Опис	Конекција на плочи
1	-	-	-
2	-	-	-
3	-	-	-
4	-	-	-
5	A+5	Неинвертована фаза А 5. енкодера	Пин 2, U8 SN65LBC173 чипа
6	B+5	Неинвертована фаза В 5. енкодера	Пин 6, U8 SN65LBC173 чипа
7	A-6	Инвертована фаза А 6. енкодера	Пин 15, U8 SN65LBC173 чипа
8	B+6	Неинвертована фаза В 6. енкодера	Пин 10, U8 SN65LBC173 чипа
9	A+3	Неинвертована фаза А 3. енкодера	Пин 2, U7 SN65LBC173 чипа
10	B+3	Неинвертована фаза В 3. енкодера	Пин 6, U7 SN65LBC173 чипа
11	A-4	Инвертована фаза А 4. енкодера	Пин 15, U7 SN65LBC173 чипа
12	B+4	Неинвертована фаза В 4. енкодера	Пин 10, U7 SN65LBC173 чипа
13	B-4	Инвертована фаза В 4. енкодера	Пин 9, U7 SN65LBC173 чипа
14	-	-	-
15	-	-	-
16	-	-	-
17	-	-	-
18	A-5	Инвертована фаза А 5. енкодера	Пин 1, U8 SN65LBC173 чипа
19	B-5	Инвертована фаза В 5. енкодера	Пин 7, U8 SN65LBC173 чипа
20	A+6	Неинвертована фаза А 6. енкодера	Пин 14, U8 SN65LBC173 чипа
21	B-6	Инвертована фаза В 6. енкодера	Пин 9, U8 SN65LBC173 чипа
22	A-3	Инвертована фаза А 3. енкодера	Пин 1, U7 SN65LBC173 чипа
23	B-3	Инвертована фаза В 3. енкодера	Пин 7, U7 SN65LBC173 чипа
24	A+4	Неинвертована фаза А 4. енкодера	Пин 14, U7 SN65LBC173 чипа
25	GND	Маса прилагодено-комуникационе картице	

Струјни капацитет за све пинове овог конектора, заправо представља струјни капацитет улаза компаратора. Ова вредност није константна, али се приближно може исказати формулом $I_{input}[mA] \approx 0.08 \frac{mA}{V} \cdot V_{input} - 0.2mA$, која важи за температуру од 25°C, напајање од +5V и када је други улаз компаратора повезан на масу.

3.2.6. Конектор JDR1

JDR1 је 37-пински конектор. На њему се налази 8 улаза за AD конверзију, 12 излаза DA конвертора, 6 PWM (*Pulse Width Modulation*) излаза, 9 пинова опште намене, напајање и маса. С обзиром да одређени елементи (отпорници и један MAX525 чип) фале на плочи, то од свих наведених оперативни су само 8 пинова за AD конверзију, и 8 пинова DA конвертора, напајање и маса.

У следећој Табела 3.7 дате су спецификације свих пинова овог конектора, без обзира да ли су они тренутно оперативни или не. За оне који нису тренутно у функцији то је наглашено блеђим бројем у колони пин. У колони конекција на плочи, дати су подаци који о вези која би постојала уколико би све компоненте биле присутне на плочи.

Са овог конектора одговарајући број пинова је изолован, односно води се на прилагодено-изолациону картицу. Бројеви пинова који се воде на прилагодено-изолациону картицу су задебљани у колони ”пин” у Табела 3.7.

Табела 3.7 Спецификације пинова конектора JDR1

Пин	Ознака	Опис	Конекција на плочи ¹	Опсег
1	Vcc	Напајање прилагодено-комуникационе картице (са GAME порта)		+5V
2	GND	Маса прилагодено-комуникационе картице (маса рачунара)		0V
3	IO7	Пин опште намене	Пин 84 FPGA чипа	-
4	IO5	Пин опште намене	Пин 82 FPGA чипа	-
5	IO3	Пин опште намене	Пин 80 FPGA чипа	-
6	IO1	Пин опште намене	Пин 78 FPGA чипа	-
7	PWM5	PWM излаз	Пин 69 FPGA чипа	-
8	PWM3	PWM излаз	Пин 67 FPGA чипа	-
9	PWM1	PWM излаз	Пин 65 FPGA чипа	-
10	ADCH6	Аналогни улаз за AD конверзију	Пин 7 MAX1202 чипа	0-4.1V
11	ADCH4	Аналогни улаз за AD конверзију	Пин 5 MAX1202 чипа	0-4.1V
12	ADCH2	Аналогни улаз за AD конверзију	Пин 3 MAX1202 чипа	0-4.1V
13	ADCH0	Аналогни улаз за AD конверзију	Пин 1 MAX1202 чипа	0-4.1V
14	DA1CH0	Аналогни излаз DA конвертора	Пин 2 и 3 U1 MAX525 чипа	0-3.3V
15	DA2CH0	Аналогни излаз DA конвертора	Пин 2 и 3 U3 MAX525 чипа	0-3.3V
16	DA3CH0	Аналогни излаз DA конвертора	Пин 2 и 3 U5 MAX525 чипа	0-3.3V
17	DA3CH2	Аналогни излаз DA конвертора	Пин 16 и 17 U5 MAX525 чипа	0-3.3V
18	DA2CH2	Аналогни излаз DA конвертора	Пин 16 и 17 U3 MAX525 чипа	0-3.3V
19	DA1CH2	Аналогни излаз DA конвертора	Пин 16 и 17 U1 MAX525 чипа	0-3.3V
20	IO9	Пин опште намене	Пин 86 FPGA чипа	-
21	IO8	Пин опште намене	Пин 85 FPGA чипа	-
22	IO6	Пин опште намене	Пин 83 FPGA чипа	-
23	IO4	Пин опште намене	Пин 81 FPGA чипа	-
24	IO2	Пин опште намене	Пин 79 FPGA чипа	-
25	PWM6	PWM излаз	Пин 70 FPGA чипа	-
26	PWM4	PWM излаз	Пин 68 FPGA чипа	-
27	PWM2	PWM излаз	Пин 66 FPGA чипа	-
28	ADCH7	Аналогни улаз за AD конверзију	Пин 8 MAX1202 чипа	0-4.1V
29	ADCH5	Аналогни улаз за AD конверзију	Пин 6 MAX1202 чипа	0-4.1V
30	ADCH3	Аналогни улаз за AD конверзију	Пин 4 MAX1202 чипа	0-4.1V
31	ADCH1	Аналогни улаз за AD конверзију	Пин 2 MAX1202 чипа	0-4.1V
32	DA1CH1	Аналогни излаз DA конвертора	Пин 4 и 5 U1 MAX525 чипа	0-3.3V
33	DA2CH1	Аналогни излаз DA конвертора	Пин 4 и 5 U3 MAX525 чипа	0-3.3V
34	DA3CH1	Аналогни излаз DA конвертора	Пин 4 и 5 U5 MAX525 чипа	0-3.3V
35	DA1CH3	Аналогни излаз DA конвертора	Пин 18 и 19 U1 MAX525 чипа	0-3.3V
36	DA2CH3	Аналогни излаз DA конвертора	Пин 18 и 19 U3 MAX525 чипа	0-3.3V
37	DA3CH3	Аналогни излаз DA конвертора	Пин 18 и 19 U5 MAX525 чипа	0-3.3V

¹ Код описа конекција на плочи занемарени су отпорници који се налазе на тој линији везе

Пинови опште намене и PWM излази, су преко отпорника (који тренутно нису залемљени на плочу), повезани на пинове FPGA чипа, тако да можемо рећи да ће струјни капацитети ових сигнала били 12mA.

3.3. Коришћене компоненте

У овом поглављу дат је преглед основних карактеристика компоненти које се налазе на прилагодено-комуникационој картици. Уз опис саме компоненте и њених могућности, назначене су оне које се користе у конкретної реализацији. Због сложености функције коју врши, то није учињено једино код програмабилног FPGA чипа, па је у овом поглављу описан само његов хардвер, а функција која је имплементирана на њему детаљно је описана у посебном прилогу (ПРИЛОГ В – Функција имплементирана на FPGA чипу).

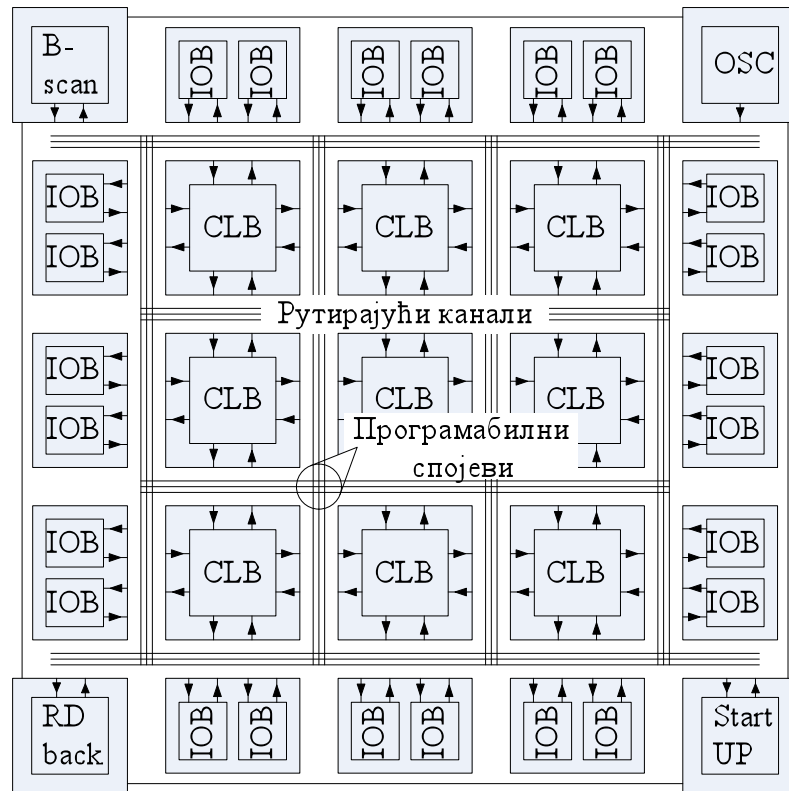
3.3.1. Xilinx® Spartan XCS20-VQ100C

FPGA (*Field-programmable gate array*) компоненте се јако често користе у развоју нових уређаја јер знатно скраћују време развоја дизајна, подржане су одговарајућим софтвером од стране самога произвођача и могу се репрограмирати готово неограничен број пута. Високе су густине паковања, доста су брзе (преко 80MHz), имају уграђен RAM на чипу, па као такве могу заменити већину специфичних - ASIC (*Application-specific integrated circuit*) чипова, а уз све то FPGA компоненте су знатно јефтиније.

У реализацији овог пројекта коришћен је чип XCS20-VQ100C произвођача Xilinx®. Коришћени чип је један од десет чипова Spartan фамилије FPGA чипова овог произвођача. Сам назив чипа садржи: тип FPGA чипа – XCS20 (*Xilinx Chip Spartan* са 20×20 конфигурабилних логичких блокова – CLB-ова), казује да је 100 пинско кућиште пластично квадратног облика (VQ – *Very thin Quad flat pack*) и да је компонента са комерцијалним температурним опсегом рада (C – *Commercial* $T_j = 0^{\circ}\text{C} - 85^{\circ}\text{C}$). Напон напајања коришћеног чипа је 5V, а потрошња у стационарном стању је око 3mA.

Структура FPGA чипа

Spartan серије FPGA чипова су регуларне, програмабилне структуре. Састоје се од конфигурабилних логичких CLB (*Configurable Logic Block*) блокова, улазно/излазних IOB (*Input/Output Block*) блокова, додатних специфичних блокова и од њихових програмабилних међу веза. Шематски приказ и распоред ових блокова на FPGA чипу дат је на Слика 3.3 [5].



Слика 3.3 Блок дијаграм FPGA чипа

CLB (Configurable Logic Block) блокови

CLB блокови се користе за имплементацију већине логичких функција на FPGA чипу. Они обезбеђују функционалну логику за имплементацију жељене логичке функције корисника. У основи, један CLB блок се састоје од 3 *look-up* табеле (LUT), које се користе као генератори произвољних логичких функција константног кашњења, 2 флип-флопа за памћење међу резултата, и две групе мултиплексера чији управљачки сигнали заправо и одређују функцију самог блока. CLB блокови су смештени у матрицу рутрајућих канала као на Слика 3.3. Функционалност сваког блока може се подесити за време конфигурисања, програмирањем интерне статичке меморије. Вредности уписане у ову меморију одређују логичку функцију и међу везе имплементиране на FPGA чипу.

Улазно/излазни блокови – IOB (*Input/Output Blocks*)

Кориснички конфигурабилни IOB блокови представљају интерфејс између пинова паковања чипа и интерне логики. Сваки IOB блок контролише један пин паковања и може га конфигурисати као улазни, излазни или бидирекциони.

Улазни бафери се могу софтверски глобално конфигурисати као TTL, или CMOS улази. И излази су такође конфигурабилни. Два глобална подешавања прага улазног сигнала и нивоа излазног напона су независна. Улазни и излазни прагови сигнала су TTL на свим конфигурираним пиновима, све док се конфигурација не унесе у чип и не специфицира како

ће они бити коришћени. У самом раду није вршена додатна спецификација тако да улази и излази FPGA чипа имају TTL напонске нивое.

Рутирајући канали

Рутирајући канали обезбеђују путање које повезују улазе и излазе CLB и IOB блокова. Сви интерни рутирајући канали су метални водови. Хоризонтални и вертикални водови секу се у блоковима који се називају програмабилне прекидачке матрице – PSM (*Programmable Switch Matrix*). Уз помоћ ових програмабилних прекидачких спојева врши се имплементација жељених путања – међу веза.

Додатни – специфични блокови FPGA чипа

Додатни специфични блокови назначени су на Слика 3.3. Види се да се физички ови блокови налазе у угловима чипа.

Осцилатор на чипу

Spartan/XL садржи интерни осцилатор на чипу. Овај осцилатор се користи за одбројавање *time-out*-а при укључивању, за брисање конфигуришуће меморије, и као извор такта – CCLK (*Configuration Clock*) у *Master* конфигурационом моду. Осцилатор ради са номиналном фреквенцијом од 8MHz, која варира са напајањем Vcc и температуром. Излазна фреквенција се налази у опсегу од 4 до 10MHz. Излаз осцилатора је опционо доступан и након конфигурације и у конкретној реализацији то се и користи са номиналном фреквенцијом од 8MHz.

Boundary Scan

Boundary scan је стандард који је увела IEEE организација (број стандарда је 1149.1). Стандард је уведен првенствено за лакше тестирање оних уређаја код којих педови нису лако доступни за мерење.

Тестирање Spartan чипа, по овом стандарду, врши се помоћу четири пина. То су 3 корисничка I/O пина (пинови 4, 5 и 6) којима су додељене специјалне функције TDI (*Test Data In*), TCK (*Test Clock*), TMS (*Test Mode Select*), и излазни пин (пин 76) коме је додељена функција *boundary scan* излаза TDO (*Test Data Output*). Према стандарду постоји и пети пин који служи да ресетује контролну логику, али код Spartana он није имплементиран.

Коришћењем ових сигнала, корисник може серијски учитавати команде и податке у чип да би контролисао излазе или пак да би читао улазе. На овај начин се такође може извршити и целокупна конфигурација Spartan чипа.

Тестирање се врши секвенцама података које се кроз чип преносе серијски од једног до другог IOB блока, који су распоређени по ободу чипа (Слика 3.3), па отуда и потиче назив стандарда – *boundary scan* (скенирање границе – обода).

Readback

Корисник може прочитати натраг (*Readback*) податке, тј. садржај конфигурационе меморије и ниво одређених интерних чворова без ометања нормалног рада уређаја.

Readback не само да даје спуштenu конфигурацију, већ може такође садржати и тренутно стање уређаја, представљено путем садржаја флип-флопова и лечева CLB-ова и IOB блокова, као и садржаје генератора функција који се користе као RAM меморија.

Startup

Startup је прелазни процес из конфигурационог у радни, кориснички, процес. *Startup* блок се може, у софтверу, укључити у пројекат укључивањем одговарајућег блока из библиотеке за сам чип. Овај блок садржи неке опције које се могу активирати у овом периоду. На пример, сигнал такта CLK који у овом прелазном процесу може бити контролисан спољашњим осцилатором, а не подразумеваним CCLK клокком; затим улазом GSR (*Global Set/Reset*) може се извршити глобални *set/reset* чиме се свим регистрима при укључењу (*startup*-у) поставља вредност на логичку „1“, односно логичку „0“. Такође, овај блок садржи и улаз GTS (*Global 3-state*) којим се експлицитно може задати да сви излазни пинови буду у стању високе импедансе. У конкретној реализацији, екстерни такт није коришћен, па је улаз CLK остао не повезан; почетно стање излаза није форсирано у стање високе импедансе, па је GTS улаз повезан на масу, али је стање свих флип-флопова (регистара, променљивих) дефинисано на почетну вредност „0“ повезивањем GSR улаза на масу (ова чињеница се користи у одељку 7.2 ПРИЛОГ В – Функција имплементирана на FPGA чипу).

Конфигурација FPGA чипа

Коришћени Spartan чип има два конфигурациона мода који се одређују стањем MODE пина (пин 24 FPGA чипа):

MODE = 1 – *Slave Serial* мод

MODE = 0 – *Master Serial* мод.

Ако се жели *Slave Serial* мод, с обзиром на уграђени *pull-up* отпорник, пин може остати неповезан док за *Master Serial* мод, овај пин треба директно повезати на масу (GND), или путем *pull-down* отпорника од 1k Ω , или мање. После конфигурације, овај пин није више у употреби.

У току конфигурације, неки од I/O пинова се привремено користе за процес конфигурације.

У реализацији пројекта коришћен је *Master Serial* мод. У овом моду користи се интерни осцилатор за генерисање CCLK (*Configuration Clock*) такта за тактовање потенцијалног *Slave* уређаја односно Xilinx серијски-конфигурационог PROM-а (SPROM-а) као што је случај у овом раду. Фреквенција CCLK такта је подесива било на 1MHz (подразумевана вредност) или на 8MHz. Конфигурација увек почиње од подразумеване

вредности такта, а затим се може променити на вишу. Толеранција фреквенције је од -50% до +25%.

Коришћење PROM-а за конфигурацију чипа је напредна идеја која се јако често користи у професионалном дизајну. Конфигурациони садржај остаје у PROM-у и када уређај није више под напајањем, а при новом укључењу се код аутоматски, серијски, спушта у чип и није потребно додатно повезивање каблова са рачунаром ради поновног програмирања чипа.

CCLK излаз чипа представља такт за CLK улаз Xilinx SPROM-а, а DATA излаз за податке SPROM-а је повезан на DIN улаз FPGA чипа. На сваку узлазну ивицу CCLK-а инкрементира се интерни адресни бројач серијског PROM-а, и врши се прихватање податка на DIN улазу чипа. Улаз $\overline{CE}$ серијског PROM-а може се повезати било на пин $\overline{LDC}$ или DONE, FPGA чипа, а у конкретној реализацији изабрана је прва опција. Такође за исправан рад потребне су још две међу везе пинова OE/ $\overline{RESET}$ и D4/CF SPROM-а, са $\overline{INIT}$ и PROGRAM пиновима FPGA чипа.

Конфигурациони подаци се преносе у низу (*beat stream*) који је за сваки уређај специфичног формата, али су низови исти и за *Master* и за *Slave Serial* мод.

У *Slave Serial* моду екстерни сигнал тактује CCLK улаз FPGA чипа (најчешће, тај екстерни сигнал долази са чипа који је конфигурисан да ради у *Master Serial* моду).

Опис пинова

На Spartan чиповима постоје три врсте пинова:

- перманентно додељени пинови
- кориснички I/O пинови који могу имати специјалне функције
- кориснички програмабилни I/O пинови.

Пре и за време конфигурације, сви излази који се не користе у процесу конфигурације су тростатички са активном I/O *pull-up* отпорном мрежом. Након конфигурације, ако се неки IOB блок не користи, он ће бити конфигурисан као улаз са I/O *pull-up* отпорном мрежом која остаје активна.

3.3.2. Xilinx-ов PROM XC18V00

Улога PROM-а у овом систему је за смештање програма за конфигурацију FPGA чипа. Коришћени PROM је Xilinx-ов XC18V512. Само име говори да се ради о Xilinx чипу (*Xilinx Chip*) типа 18v, са 512 kb PROM меморије, а изабрано кућиште је квадратно 20 пинско (PC20). Као што се види коришћени PROM је од истог произвођача као и сам FPGA чип, односно усклађен је са њим према упутству. Конфигурација FPGA чипа, односно пренос података из PROM-а у FPGA чип се врши серијски, па се сам PROM, са серијским начином преноса често назива и SPROM (*Serial PROM*).

PROM чип се напаја напоном од +3.3V док се његово програмирање (упис) врши стандардним TTL напонским нивоима, односно са +5V. Такође сви његови улази/излази су у потпуности компатибилни са уређајима који раде на +5V. Потрошња чипа у активном режиму не прелази 25mA, док у *standby* моду није већа од 10mA. Температура на којој се могу очувати уписани подаци је од -40°C до +85°C. Чип подржава IEEE 1149.1 *boundary-scan* (JTAG) стандард. Према FPGA чипу има једноставан серијски/паралелни интерфејс. У конкретном случају FPGA чип је конфигурисан да ради у *Master Serial* моду, па је за конфигурацију коришћен основни серијски интерфејс за пренос података ка FPGA чипу путем излазне линије D0 (пин 1), која је повезана на улазни пин DIN, FPGA чипа (пин 72). Паралелни конфигурациони мод може се подесити коришћењем одговарајућег софтвера за програмирање PROM-а и уписом одговарајућег бита у његов контролни регистар. Уколико су сигнали $\overline{CE}$ и OE активни, подаци ће се преносити на сваку узлазну ивицу такта CCLK (пин 74) који се са FPGA чипа доводи на CLK (пин 3) PROM-а. Уколико су ови сигнали неактивни, тада се PROM налази у *standby* моду. $\overline{CF}$ линија PROM-а (пин 7) је повезана на $\overline{PROGRAM}$ (пин 52) FPGA чипа и омогућава његово конфигурисање и док је FPGA чип у раду.

Програмирање PROM-а се може вршити било док је чип у систему, или екстерно путем програматора. Програмирање PROM-а се врши путем стандардног 4-пинског JTAG прикључка, односно путем линија TCK (*Test Clock*), TMS (*Test Mode Select*), TDI (*Test Data In*) и TDO (*Test Data Output*) које представљају редом такт, мод, серијски улаз и серијски излаз. Инструкцијски регистар XC18V чипа има 8 бита и налази се између TDI и TDO линија.

С обзиром да FPGA чип ради у *Master Serial* моду, при укључењу напајања врши се аутоматско уписивање података (конфигурационог програма) из PROM-а у FPGA чип, односно врши се конфигурација FPGA чипа. Након конфигурације FPGA чипа, улога PROM-а у колу више није битна, па на први поглед и његова улога није толико битна. Међутим, уочимо да ће FPGA чип остати конфигурисан само док је под напоном, а при сваком новом укључењу морали би вршити на неки други начин конфигурацију, а сваки други начин захтевао би поновно генерисање конфигурационог програма, поновно повезивање са рачунаром, и поновни упис података у FPGA чип. Дакле очигледно је да је коришћење PROM-а знатно боље и елегантније решење.

3.3.3. MAX525

MAX525 чип је 12-битни D/A конвертор, ниске потрошње, са серијским уписом података и са 4 аналогна излаза. Налази се у 20-пинском DIP кућишту. Напаја се са једностраним напајањем од +5V. Просечна потрошња му је око 0.85mA. Потрошња се може смањити коришћењем софтверских и хардверских *power-down* модова, али у нашем случају они се не користе (пин PDL је повезан на напајање (+5V)), пошто потрошња овог чипа није критична. Сви улази чипа су TTL/CMOS компатибилни.

Путем 3 пина који представљају серијски интерфејс овог чипа, MAX525 прима команду са податком који треба конвертовати, и то смешта у 16-битни *shift* регистар. Овај серијски пренос се врши путем DIN линије за серијски пренос података и на узлазну ивицу серијског SCLK (*Serial Clock*) блока који долазе са кориснички дефинисаних пинова FPGA чипа. Такође неопходан сигнал за селекцију жељеног чипа је $\overline{CS}$, и пренос је валидан само за време док је овај сигнал на логички ниском нивоу. Такође на чипу се налази и серијски излаз

DOUT који није коришћен, а који би се могао користити за проверу преноса података или при уланчавању више MAX525 чипова.

Команда садржи 2 адресна и 2 контролна бита. Сваки D/A конвертор има двоструко баферисани улаз, који се састоји од улазног и DAC регистра. Учитани 16-битни податак се смешта у сва 4 улазна регистра. Освежавање новим вредностима DAC регистра се може извршити по на особ или одједном уписом исте вредности у сва 4 регистра, у зависности од задате команде. Задата команда такође може имати и функцију слања чипа у *power-down* мод, или задавања вредности на кориснички дефинисаном пину UPO итд., али ове функције нису коришћене у самом раду. Серијски интерфејс је компатибилан са SPI/QSPI и MICROWIRE стандардима преноса.

На излазу из чипа се налазе појачавачи, чији су негативни улази доступни кориснику за прављење произвољног појачања напона на излазу. У конкретној реализацији ти пинови су спојени са излазима појачавача и на тај начин је постигнуто јединично појачање – јединични бафери. Излазни напон је пропорционалан примењеној напонској референци. MAX525 садржи две различите напонске референце за А и В D/A конверторе, и за С и D D/A конверторе. У самој реализацији улази за задавање референци REFAB и REFCD су међусобно спојени и спојени на напајање од 3.3V. С обзиром на то, и максимална вредност излазног аналогног напона, за задату дигиталну вредност 111hex (4095dec), износи 3.3V.

3.3.4. MAX1202

MAX1202 је чип са осам 12-битних A/D конвертора. Налази се у 20-пинском DIP кућишту. Може се напајати или се једностраним напајањем од +5V, или са симетричним напајањем $\pm 5V$. Дизајниран је тако да има комбиноване напоне од +5V аналогног напона, док дигитални део кола ради на +3V, међутим дигитални излази се могу подесити на жељени напонски ниво повезивањем VL улаза на одговарајући напонски ниво од рецимо стандарних +5V, што је и учињено при реализацији.

Чип има осам канала за A/D конверзију који су мултиплексирани прати/памти колом широког пропусног опсега (133kHz). Осам сингл улаза се могу конфигурисати и тако да чип ради као A/D конвертор за четири диференцијална аналогна улаза. Минимално време потребно за аквизицију сигнала је 1.5 μ s. Вредност улазног аналогног сигнала за исправну конверзију мора се налазити у опсегу од Vss-50mV до Vdd+50mV, док су крајње вредности исправног рада чипа одређене заштитним диодама и износе Vss-0.3V и Vdd+0.3V. Саму конверзију врши капацитивни A/D конвертор са сукцесивним апроксимацијама. MAX1202 има уграђену референцу напона од 4.096V, а може се користити и спољашња референца напона, која овде није потребна, па су улаз REFADJ и излаз REF спојени одговарајућим кондензаторима на масу. На овај начин је управо дефинисана и максимална вредност аналогног улазног напона. Дакле, за A/D конверзију аналогног напона од 4.096V добија се дигитална вредност 111hex (4095dec).

Посебно добра одлика овог чипа је ниска потрошња која се чак може смањити слањем чипа у *power-down* мод, помоћу пина SHDN, у периодима између конверзија. Међутим како потрошња овог чипа у целом нашем систему није критична, то се никада не ради, и овај пин је залемљен на Vss. Просечна потрошња чипа је око 1.5mA.

Интерфејс за слање дигиталних конвертованих вредности је јако једноставан серијски, са 4 линије за пренос и линије такта. Директно се може повезивати на SPI и *Microwire* уређаје. Брзина серијског преноса података је до 2MHz. Сама конверзија и пренос података отпочињу серијским уписом контролне речи за отпочињање конверзије, путем DIN линије, у контролни бајт. Контролна реч почиње са јединицом, као старт битом, а затим следи 3-битни редни број A/D улаза, за који се жели извршити конверзија. Преостала 4 бита одређују да ли је униполарна или биполарна конверзија, да ли су сингл или диференцијални улази, и која врста блока односно *power-down* мода се користи. У конкретном случају конверзија је униполарна, сингл су улази и чип ради у *External Clock* моду, тако да је стандардни формат командне речи за отпочињање конверзије „1XXX 1111“ (где је „XXX“ бинарно записани редни број улаза чија се A/D конверзија врши). Иначе само слање података је могуће тек након обарања линије $\overline{CS}$ на низак ниво. Такт чипу задаје FPGA чип путем пина опште намене који се доводи на SCLK (пин 19) MAX1202 чипа. При слању прве командне речи, аутоматски се и чита претходних 8 бита из контролног регистра путем DOUT линије, али се тај први бајт игнорише. Потом се заредом шаљу два бајта са нулама, а читају се резултати који представљају 12-битни резултат конверзије. Након извршеног преноса линија $\overline{CS}$ се поново враћа на висок ниво. Иначе, треба напоменути да се сама аквизиција врши при слању последња три бита прве командне речи, док се сама конверзија, у *External Clock* моду, врши истовремено са самим преносом података, с обзиром да је конверзија са сукцесивним апроксимацијама, па се резултат добија бит по бит почев од MSB до LSB бита, и у том редоследу се бити и шаљу. Поред поменутих сигнала за пренос података (SCLK, DIN, DOUT и $\overline{CS}$), за пренос се користи и SSTRB излазни сигнал. Пре и после конверзије он се налази у стању високе импедансе. У *External Clock* моду, за време конверзије, SSTRB је на ниском нивоу за време преноса прве контролне речи, затим се подиже на висок ниво у коме остаје једну периоду такта док A/D конвертор не донесе одлуку о MSB биту, а затим се редом доносе одлуке о следећим битовима и истовремено врши пренос. Поред *External Clock* мода, постоји и *Internal Clock* мод, а основна разлика је у томе што у *External Clock* моду, екстерни клок поред тога што се користи за пренос података (клок за DIN и DOUT), користи се и за драјвовање корака A/D конверзије. *External Clock* мод више задржава сам процесор (FPGA чип), али је конверзија бржа, па је стога и донета одлука о коришћењу овога мода.

3.3.5. SN65LBC173

SN65LBC173 је четвороструки диференцијални линијски пријемник са тростатичким излазом. Погодан је за пријем података у шумовитим срединама са дугим линијама за пренос, и задовољава захтеве ANSI стандарда и ITU препорука. У раду је коришћен ANSI RS-485 стандард за пренос података. Не препоручује се за веће брзине преноса од 10Mb/s.

Чип је ниске потрошње која не прелази 20mA. Напон напајања чипа је +5V. Коришћени чип се налази у 16-пинском DIP кућишту, а радни температурни опсег му је од -40°C до +85°C.

Чип садржи четири диференцијална улаза, односно четири пријемника који имају заједнички сигнал дозволе, који представља излаз „или“ кола са два улаза од којих је један активан на ниском а други на високом нивоу. Ови пинови дозволе (пин 4 и пин 12) су повезани тако (на Vcc и на GND, редом) да су сви излази увек активни. Сваки пријемник је заправо хистерезисни компаратор, чија ширина хистерезиса износи $\pm 200\text{mV}$. Сами улази се

могу налазити у опсегу од $-7V$ до $+12V$, али њихова међусобна разлика не сме бити већа од $6V$, по апсолутној вредности.

Постојање хистерезиса служи да отклони шум. У конкретној реализацији се са енкодера, према RS-485 стандарду враћају одговарајући сигнал и његов комплемент. Довођењем таквих сигнала на улазе пријемног компаратора, на његовом излазу добићемо сигнал идентичан оригиналном, али са отклоњеним шумом, што нам је и био циљ.

3.3.6. LM317

LM317 је напонски регулатор који обезбеђује, на плочи, напајање од $+3.3V$. То је DC/DC конвертор који од улазног једносмерног напона од $+5V$, прави стабилан излазни једносмерни напон од $+3.3V$. Излазни напон се подешава помоћу два отпорника, и може се налазити у опсегу од 1.2 до $37V$, а максимална разлика улазног и излазног напона може износити до $40V$. Струјни капацитет излаза је до $1.5A$. Може се наћи у више кућишта, а коришћено је пластично 3-пинско TO-220 кућиште. Радни температурни опсег чипа је од 0 до $125^{\circ}C$.

4. ПРИЛАГОДНО-ИЗОЛАЦИОНА КАРТИЦА

Пројектовање изолационог дела или прилагодено изолационе картице представља централни део дипломског рада. Она је ново направљена картица додата у систем, и физички се налази „приљубљена“ уз прилагодено-комуникациону картицу.

Галванска изолација сигнала је неопходан поступак у сваком дизајну где постоји могућност пренапона и негативног утицаја једног блока уређаја на други. Задатак ове картице је да изврши усклађивање напонских нивоа и галванску изолацију сигнала између прилагодено-комуникационе картице и одговарајућих сигнала са лабораторијске станице „Вектра“.

Како постоје и аналогни и дигитални сигнали које је требало пренети и чију је галванску изолацију требало остварити, и како се одређени преносе од „рачунара ка мотору“, док они други путују од „мотора ка рачунару“ то је ово поглавље подељено у 4 целине које обухватају све претходно поменуте случајеве изолација.

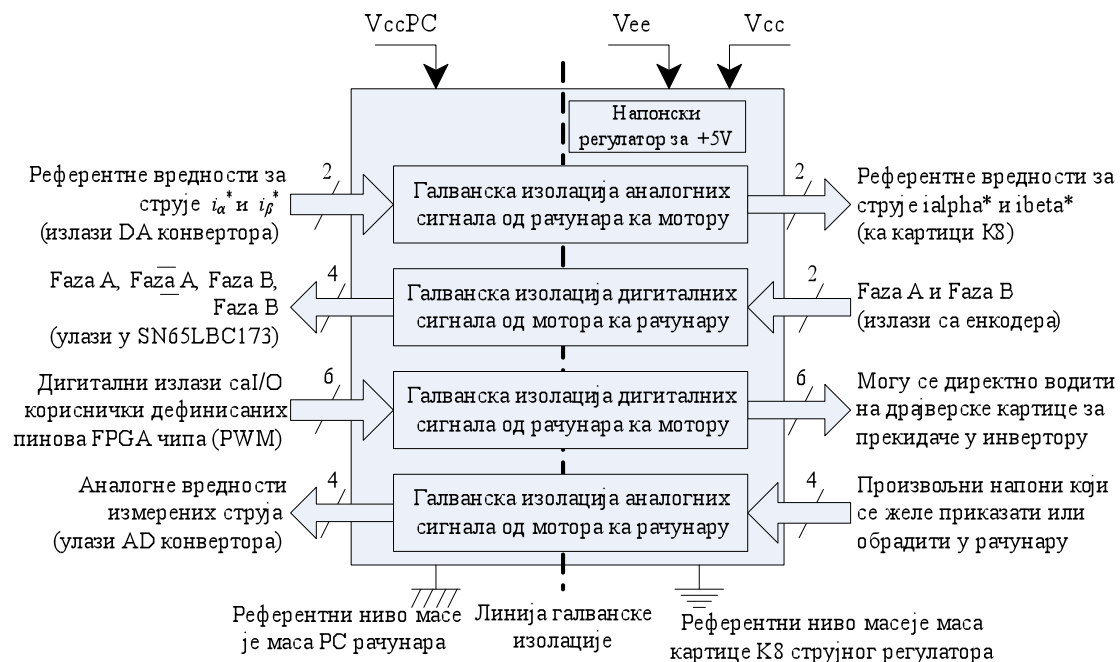
Реализоване изолације и прилагођења сигнала, које је било потребно имплементирати су:

- Аналогна изолација сигнала од рачунара ка мотору² (за изолацију задатих референтних вредности за струје статора у $\alpha\beta$ координатном систему - i_{α}^* и i_{β}^*)
- Аналогна изолација сигнала од мотора ка рачунару (за изолацију мерених струја у фазама мотора, које ће се враћати као повратна информација, за алгоритам када се мотором буде управљало помоћу PWM импулса)
- Дигитална изолација сигнала од рачунара ка мотору (за изолацију PWM сигнала који ће директно преко драјвера палити прекидаче инвертора)
- Дигитална изолација сигнала од мотора ка рачунару (за изолацију сигнала фаза енкодера)

² термини изолација сигнала од рачунара ка мотору, и од мотора ка рачунару, нису баш најпрецизнији, пошто они заправо изолују сигнале између прилагодено-комуникационе картице и управљачког погона мотора, али ће се без обзира на то, с обзиром на јасноћу смера тока сигнала, ови изрази користити и у наставку текста

На крају поглавља дат је и *Layout* реализоване прилагодно-изолационе картице, а потом и фотографија финалног производа.

Да би се лакше сагледао проблем сигнала које треба изоловати и доступних напајања, на следећој слици (Слика 4.1) приказан је интерфејс прилагодно-изолационе картице.



Слика 4.1 Блок шема прилагодно-изолационе картице

4.1. Изолација аналогних сигнала од рачунара ка мотору

Аналогни сигнали које треба изоловати су референтне вредности за струје статора у $\alpha\beta$ координатном систему. Ти сигнали су синусоиде одговарајуће амплитуде, и одговарајуће учестаности, и задају се путем рачунара. Рачунар податке (одбирке синусоиде) прослеђује ка мотору путем прилагодно-комуникационе картице, која врши D/A конверзију ових сигнала, затим се аналогни сигнали галвански изолују и напонски усклађују на прилагодно-изолационој картици, одакле се сигнали воде на место задавања референтних вредности на картици 8, лабораторијске станице "Вектра". Ови сигнали, референтних вредности, су у некадашњој реализацији долазили са, сада избачене, прилагодне картице 2.

Амплитуда се на рачунару задаје као 12-битни број у децималном облику, тако да је опсег за њено задавање од 0 до 4095. У случају задавања веће или мање вредности, након конверзије децималног у бинарни запис, на самој прилагодно-комуникационој картици доћи ће до одсецања виших бита, и конвертоваће се само 12-битни податак. Конверзију врши D/A конвертор – чип MAX525, чији је максимални излазни напон подешен улазима за задавање референци REFAB и REFCД на 3.3V. Дакле дигитално задата вредност одбирке синусоиде, претвара се у напонски сигнал чија је вредност од 0 до 3.3V. Постојећи хардвер

лабораторијске станице (картица 8), има као улазне параметре струје i_{α}^* и i_{β}^* , али су напонски нивои, који се овде захтевају, за ове аналогне напоне, од -10V до +10V. То значи да је задатак прилагодено-изолационе картице, да поред изолације сигнала, изврши и транслирање нивоа напона тог сигнала. Ако улазе у ову картицу, који су напонски нивои, иако представљају референце за струје, обележимо са V_{ain} , а излазе са V_{aout} , онда ће формула преноса прилагодено изолационе картице, с обзиром на опсеге у којима се могу наћи улаз и излаз ($V_{ain} \in [0 - 3.3V]$, $V_{aout} \in [-10V - +10V]$) бити:

$$V_{aout} = 6.06 \cdot V_{ain} - 10V.$$

Основни циљ реализоване структуре, дакле и јесте да што боље задовољи претходну формулу, односно да се постигне што већа линеарност. Такође поред овога, јако битан параметар је свакако и пропусни опсег изолационог блока. Сходно примени, да ови сигнали представљају референтне струје статора, можемо закључити да је потребна учестаност задатих синусоида реда фреквенције мреже, односно око 50Hz. Стога је као услов постављен 3dB пропусни опсег од око 1kHz, односно још строжији услов, да слабење амплитуде не буде веће од 10% до поменуте учестаности од 1kHz.

Прегледности ради, конекције пинова и напонски опсези, улаза и излаза, шеме која врши аналогну изолацију дате су следећом табелом (Табела 4.1).

Табела 4.1 Карактеристике аналогних изолованих сигнала, од рачунара ка мотору

	Страна до рачунара (референтни ниво масе је маса рачунара)		Страна до мотора (референтна маса је K2-GND)	
	Пин JDR1 37 конектора п-к картице	Напонски опсег	Пин конектора J1 п-и картице	Напонски опсег
i_{α}^*	Пин 14 (DA1CH0)	0 – 3.3V	Пин 13	-10 – +10V
i_{β}^*	Пин 16 (DA3CH0)		Пин 25	

п-к картице – прилагодено-комуникационе картице

п-и картице – прилагодено-изолационе картице

Шема која врши аналогну изолацију, и задовољава услове из табеле, може се реализовати на више начина.

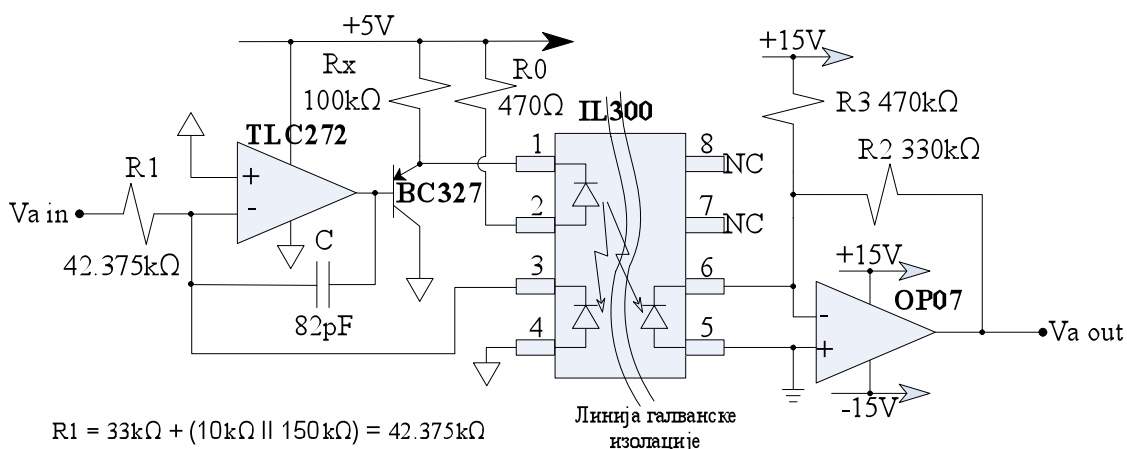
За проблем решавања изолације аналогних сигнала, у самом раду, анализирана су два решења:

- 1) Изолација коришћењем линеарног оптокаплера IL300, и
- 2) Изолација коришћењем изолационог појачавача HCPL-7840

Сада ћемо рећи нешто више о свакој од ових реализација.

4.1.1. Изолација аналогних сигнала коришћењем линеарног оптокаплера IL300

Прва идеја изолације аналогних сигнала реализована је помоћу линеарног оптокаплера IL300, одговарајућих операционих појачавача и осталих додатних компоненти. Конкретна, тестирана реализација приказана је на следећој слици (Слика 4.2).



Слика 4.2 Шема изолације аналогних сигнала коришћењем линеарног оптокаплера IL300

Као што се види линеарни оптокаплер се састоји од AlGaAs IR LED диоде, и за разлику од обичног оптокаплера, садржи раздвојену PIN фотодиоду. Раздвојена фотодиода се може посматрати као две "идентичне" фотодиоде у смислу да обе реагују на светлосни флуks са генерисањем „исте“ струје. Једна фотодиода служи за пренос сигнала ка другом делу кола који треба да је галвански изолован од дела који напаја LED диоду, док се друга фотодиода користи као информација за струје које се генеришу, и да враћа повратну информацију улазном делу о сигналу који се преноси кроз другу, њему упарену фотодиоду ка излазу.

Првобитна реализација није укључивала приказани PNP транзистор, али се јављао проблем нелинеарности при мерењу, услед тога што улазна LED диода линеарног оптокаплера није радила у линеарном делу преносне карактеристике. Разлог за то је био што операциони појачавач TLC272 није био у стању да обезбеди потребну струју која би радну тачку LED диоде могла да постави у жељени, линеарни опсег. При решавању овог проблема покушало се са тражењем операционог појачавача који има већи излазни струјни капацитет од предложеног, али с обзиром да се коришћени појачавач морао напајати са релативно ниским напоном од +5V, за жељену излазну струју, то се убацивање PNP транзистора и отпорника Rx, за конструкцију струјног бафера, испоставило као боље и знатно једноставније решење. Наравно, могућности реализације овог решења свакако је ишла у прилог чињеница да струја бафера не треба да мења смер.

Функција преноса приказане шеме, може се једноставно извести, и дата је формулом:

$$V_{aOUT} = K_3 \cdot R_2 / R_1 \cdot V_{aIN} - 15V \cdot R_2 / R_3$$

где је параметар K_3 , параметар који представља однос, количника струје која се пресликава у излазну фотодиоду у односу на струју LED диоде, и количника струје која се пресликава у струју повратне фотодиоде у односу на струју LED диоде. Једноставније формулом изражено K_3 је дато са:

$$K_3 = \frac{I_{phOUT} / I_{LED}}{I_{phFB} / I_{LED}} = \frac{I_{phOUT}}{I_{phFB}} \approx 1$$

Као што се види, и као што се и очекивало, параметар K_3 приближно има вредност 1, и зависи од упарености раздељене фотодиоде.

За потребне опсеге напона, $V_{aIN} \in [0 - 3.3V]$, и $V_{aOUT} \in [-10 - +10V]$ добија се: $K_3 \cdot R_2 / R_1 = 6.0606$ и $R_2 / R_3 = 2/3$, односно жељена функција преноса износи:

$$V_{aOUT} = 6.0606 \cdot V_{aIN} - 10V.$$

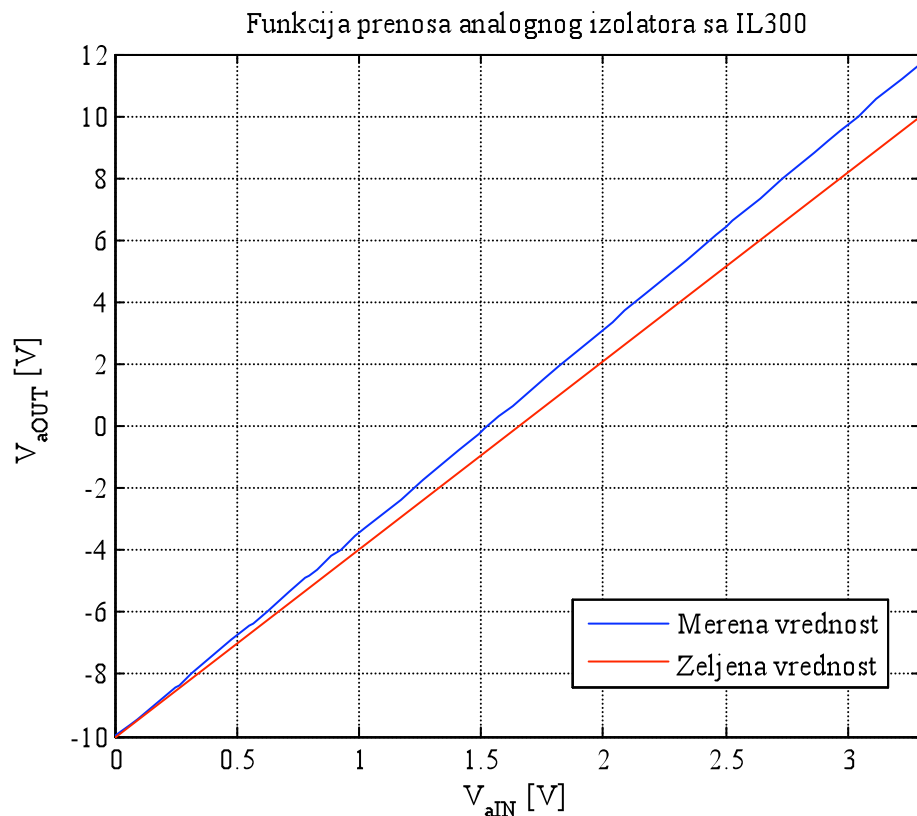
На основу добијених резултата за односе између отпорника, на основу жеље за потребном струјом да LED диода ради у линеарном режиму, и на основу податка о струјном појачању $I_{ph} / I_{LED} \approx 0.007$ (уз претпоставку да је $K_3 = 1$) дошло се до приближних вредности за отпорнике које су дате уз сваки елемент на Слика 4.2 (сви отпорници имају стандардне вредности).

За изабране вредности отпорника се добија да је функција преноса приближно:

$$V_{aOUT} = K_3 \cdot 330k\Omega / 42.375k\Omega \cdot V_{aIN} - 15 \cdot 330k\Omega / 470k\Omega$$

$$V_{aOUT} = K_3 \cdot 7.88 \cdot V_{aIN} - 10.532V.$$

Резултати мерења функције преноса, кола реализованог на протоборду у лабораторији, приказани су на следећој слици (Слика 4.3).



Слика 4.3 Функција преноса кола за изолацију аналогних сигнала са линеарним оптокаплером IL300

Са слике се може уочити висока линеарност коју даје изабрана реализација, али и одступање коефицијента правца праве, које је директна последица избора отпорника из сета стандардних вредности. Ова чињеница не представља проблем с обзиром да улазни напон можемо софтверски прескалирати, слањем пропорционално умањене вредности броја за AD конверзију.

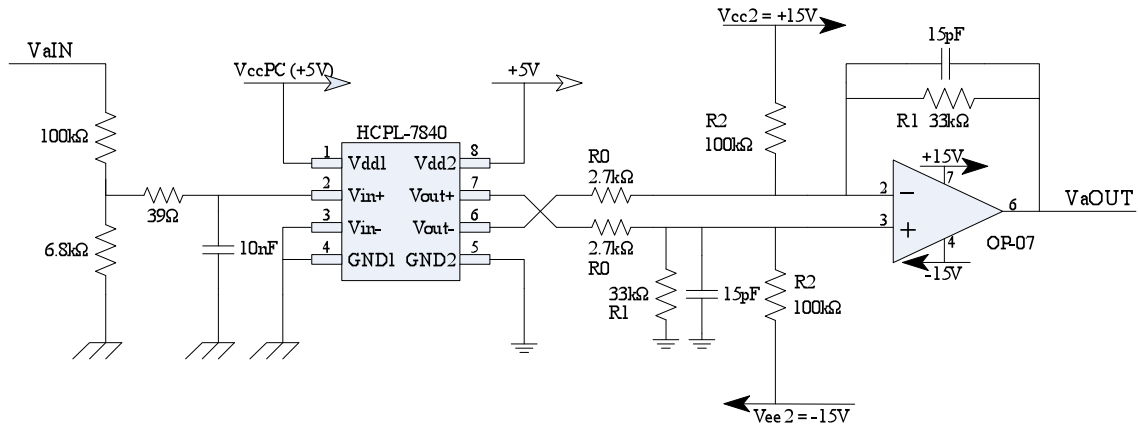
Приказано решење има висок степен линеарности. Максимално одступање од праве линије на пуном опсегу излазног напона од 21.91V, износи само 0.1238V, тако да је постигнута линеарност $0.1238V / 21.91 = 0.0056 = 0.56\%$. Пропусни опсег реализоване структуре је задовољавајући с обзиром да вредност амплитуде не пада испод 90% номиналне вредности и до фреквенције од 3kHz, а да је усвојена захтевана вредност 1kHz. Занимљиво је да на вишим фреквенцијама не долази до слабљења амплитуде већ до изобличења сигнала.

Међутим, без обзира на очигледно добре карактеристике приказане реализације, коришћена компонента IL300 има једну јако лошу особину, а то је да њене карактеристике деградирају са временом, те би с времена на време било потребно поновно подешавање отпорника, или евентуално промена параметра пропорционалности у коду, па се због тога и одустало од оваквог начина реализације аналогне изолације.

4.1.2. Изолација аналогних сигнала коришћењем изолационог појачавача HCPL-7840

Изолација аналогних сигнала помоћу изолационог појачавача HCPL-7840 даје приближно исте резултате као и претходно описана шема, али је ова компонента знатно постојанија у времену, те се очекује да додатна подешавања неће бити потребна.

Шема која остварује изолацију аналогних сигнала, усклађивање и транслирање напонских нивоа, приказана је на следећој слици (Слика 4.4).



Слика 4.4 Шема изолације аналогних сигнала од рачунара ка мотору, коришћењем изолационог појачавача HCPL-7840

Са слике се види да је са стране мотора потребно и напајање од +5V које је добијено коришћењем напонског регулатора LM78L05, који од напона напајања картице 8 ($V_{CC2}=+15V$), прави жељени напон од +5V. На шеми, нису уцртани *bypass* кондензатори, преко којих се доводи свако напајање на сваки од коришћених чипова. Коришћене вредности за ове кондензаторе су 100nF.

На самој слици обратити пажњу на различите нивое маса (самим тим и различите ознаке) са леве и десне стране изолационог појачавача HCPL-7840, што и јесте суштина ове као и свих других шема из овог поглавља.

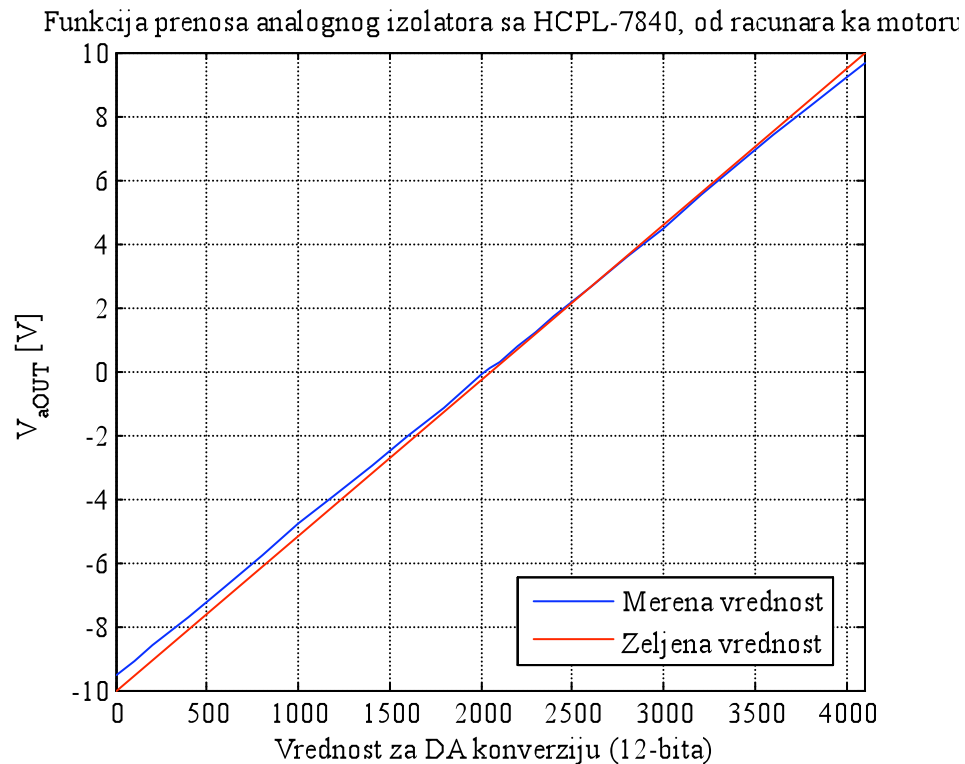
Функција преноса приказане структуре једносавно се може извести и дата је следећим изразом:

$$V_{aOUT} = \frac{R_1}{R_0} (V_{out}^+ - V_{out}^-) - \frac{R_1}{R_2} (V_{CC2} - V_{EE2}).$$

Како смо желели да што боље искористимо део линеарног опсега линеарног појачавача HCPL7840, који за улазни диференцијални напон износи од -200mV до +200mV, то се на улазу кола налази напонски разделник који улазни напон од 0 до +3.3V своди у опсег од приближно 0 до +210mV. Појачање компоненте у овом опсегу износи око 8, тако да можемо приближно рећи да ће се вредност $V_{out}^+ - V_{out}^-$ кретати у опсегу од 0V до око

$8 \cdot 210mV = 1.68V$. Овај податак као и подаци о вредностима за напајања ($V_{CC2} = +15V$ и $V_{EE2} = -15V$) дају нам могућност да извршимо одабир отпорника тако да се излазни напон креће у захтеваном опсегу од $-10V$ до $+10V$. Изабране вредности отпорника (све вредности су из стандардног сета отпорности) су приказане на самој шеми (Слика 4.4).

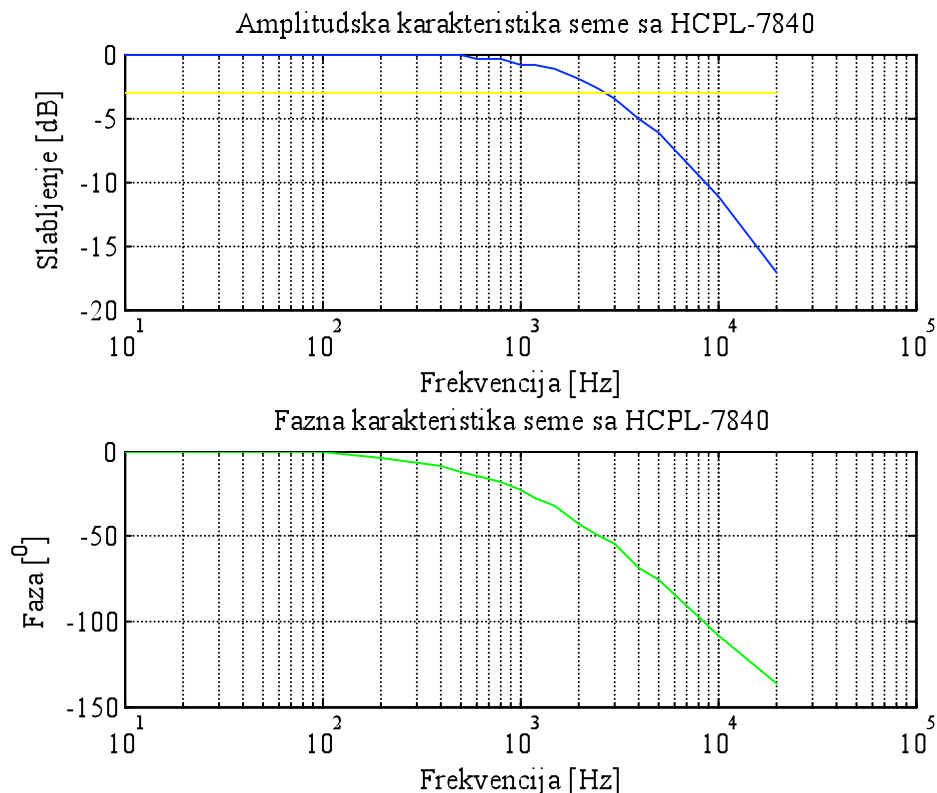
Резултати мерења функције преноса, реализованог кола, приказани су на следећој слици (Слика 4.5).



Слика 4.5 Функција преноса кола за изолацију аналогних сигнала од рачунара ка мотору, са линеарним изолатором HCPL-7840

За разлику од мерења из претходне реализације, уочава се да се овде на апциси налази дигитална вредност броја који се конвертује. То наравно суштински ништа не мења с обзиром на високу линеарност самог DA конвертора. Са слике се може уочити висока линеарност коју даје изабрана реализација, која је још и боља од претходне. Такође, захваљујући боље уклопљеним вредностима отпорности, и статичка грешка је доста мања. Максимално одступање од праве линије повучене кроз прву и последњу тачку мерења на пуном опсегу излазног напона од $19.2V$, износи само $0.096V$, тако да је постигнута линеарност $0.096V / 19.2V = 0.005 = 0.5\%$.

Ново реализована шема има нешто лошији пропусни опсег у односу на претходну. Амплитудска и фазна карактеристика дате су на следећој слици (Слика 4.6).



Слика 4.6 Фреквенцијске карактеристике реализоване шеме за изолацију аналогних сигнала од мотора ка рачунару

Са слике се може видети да је -3dB пропусни опсег око 2.75kHz . Како смо и код претходне реализације разматрали још строжији услов, да је слабљење мање од 10%, номиналне вредности то ћемо и овде навести исти податак који сада износи око 1300Hz . Напоменимо то да су амплитудска и фазна карактеристика снимљене у лабораторисјким условима одвојено од рачунара и мотора, како би без бојазни биле преспојене GND-PC и K2-GND, што је неопходно да би се помоћу једног осцилоскопа, са две сонде које имају исту референтну масу, могло извршити мерење.

Међутим ови подаци нису од великог значаја у практичном раду рачунара. Наиме рачунар задаје одбирке синусоиде, тако да са рецимо порастом фреквенције, неће доћи до слабљења амплитуде излазног сигнала, већ ће у периоди бити само мањи број одбирака. Међутим како се реализована изолација може користити и за пренос аналогних сигнала, подаци о амплитудској и фазној карактеристици, свакако су од значаја и свакако карактеришу квалитет реализоване структуре.

4.2. Изолација аналогних сигнала од мотора ка рачунару

Аналогна изолација сигнала од мотора ка рачунару представља проширење могућности лабораторијске станице "Вектра". Првенствено, она ће се користити за изолацију

мерених струја у фазама мотора, које ће се враћати као повратна информација, за алгоритам када се мотором буде управљало помоћу PWM импулса.

Мерене струје су синусоиде одговарајуће амплитуде, и одговарајуће учестаности. Сходно карактеристици мотора и номиналној струји од 2А, претпоставићемо да амплитуда струје никада не прелази 5А, а сигурности ради, приликом пројектовања изолације претпоставићемо да та вредност не прелази 7V, уколико се податак о струји мери са шантова од 1Ω. Треба уочити да се овде ради о променљивом синусоидалном напону тако да се напон на улазу може наћи у опсегу од усвојених -7V до +7V. Излани изоловани напон се води на, прилагодено-комуникациону картицу, на улаз AD конвертора MAX1202 чипа, тако да опсег изолованог напона мора бити у интервалу од 0 до +4.1V. Такође уочимо и то да пропусни опсег овог изолатора не треба да буде пуно већи од номиналне фреквенције струје, односно од 50Hz (фреквенција која би се имала уколико би се мотор директно напајао из мреже), али ћемо се, као и у претходном случају пројектовања изолације, потрудити да пропусни опсег буде већи од 1kHz.

Сходно напонским нивоима сигнала које желимо изоловати ($V_{ain} \in [-7V - +7V]$, $V_{aout} \in [0 - +4.1V]$), можемо написати функцију преноса коју реализовано коло треба да задовољава:

$$V_{aOUT} = 0.293 \cdot V_{aIN} + 2.05V .$$

Основни циљ реализоване структуре је да што боље задовољи претходну формулу, да се постигне што већа линеарност, и да се оствари што је могуће већи пропусни опсег.

Прегледности ради, конекције пинова и напонски опсези, улаза и излаза, шеме која врши аналогу изолацију дате су следећом табелом (Табела 4.2).

Табела 4.2 Карактеристике аналогних изолованих сигнала, од мотора ка рачунару

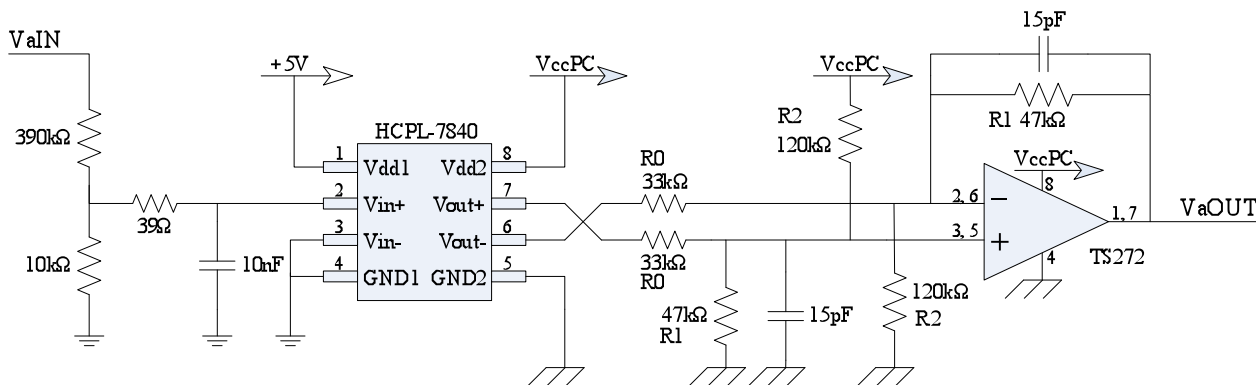
Страна до рачунара (референтни ниво масе је маса рачунара)		Страна до мотора (референтна маса је K2-GND)	
Пин JDR1 37 конектора п-к картице	Напонски опсег	Пин конектора п-и картице	Напонски опсег
Пин 12 (ADCH2)	0 – 4.1V	Још није специфицирано. За сада ово постоји само као могућност	-7V – +7V
Пин 13 (ADCH0)			
Пин 30 (ADCH3)			
Пин 31 (ADCH1)			

п-к картице – прилагодено-комуникационе картице

п-и картице –прилагодено-изолационе картице

Што се тиче броја оваквих изолација које је било потребно реализовати, одлучило се да то буде 4. Разлог за то се налази у изоловању, или на потенцијална два система по две мерене струје, или на једном систему у изоловању две или све три мерене фазне струје. У сваком случају, уколико се не искористе све четири могуће изолације, преостала једна или две се могу користити као ”осцилоскопи” за посматрање произвољних сигнала са картице K8, односно оних сигнала чија је референтна маса K2-GND, а напонски опсег од -7V до +7V.

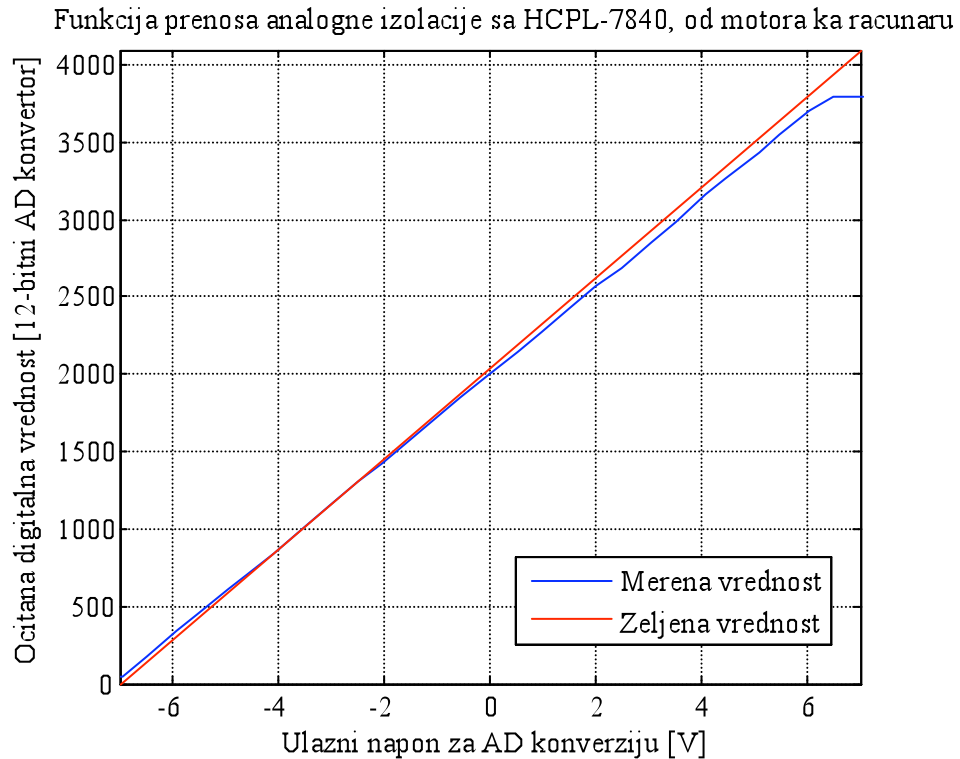
Шема која реализује жељену функцију и остварује потребну изолацију приказана је на следећој слици (Слика 4.7).



Слика 4.7 Шема изолације аналогних сигнала, од мотора ка рачунару, коришћењем изолационог појачавача HCPL-7840

Види се да је ова шема јако слична претходно описаној шеми за изолацију аналогних сигнала од рачунара ка мотору (Слика 4.4), с тим што треба приметити да је сада улазни напон симетричан, а да је излазни униполаран. Захваљујући ”добрим” карактеристикама компоненте HCPL-7840, и могућности да се изолација и линеарно појачање сигнала изврше док је улазни напон у опсегу до -200mV до +200mV, без обзира на једнострано напајање (0V и +5V), ова шема је директно добијена из претходне само заменом вредности отпорника. Такође види се да у излазном делу шеме (излазни део је сада са стране до рачунара, тј. прилагодено-комуникационе картице), поново постоје отпорници за транслирање напона који сада служе да за измерену нулту вредност струје са стране мотора подесе излазни напон на половину максималне вредности коју може да прихвати AD конвертор, тј. на $4.1V / 2 = 2.05V$. Такође, види се да се напон напајања од +5V HCPL-7840 чипа, са стране мотора морао обезбедити помоћу напонског регулатора LM78L05. На шеми, као и на претходној, нису уцртани *bypass* кондензатори, преко којих се доводи свако напајање на сваки од коришћених чипова. Коришћене вредности за ове кондензаторе су 100nF.

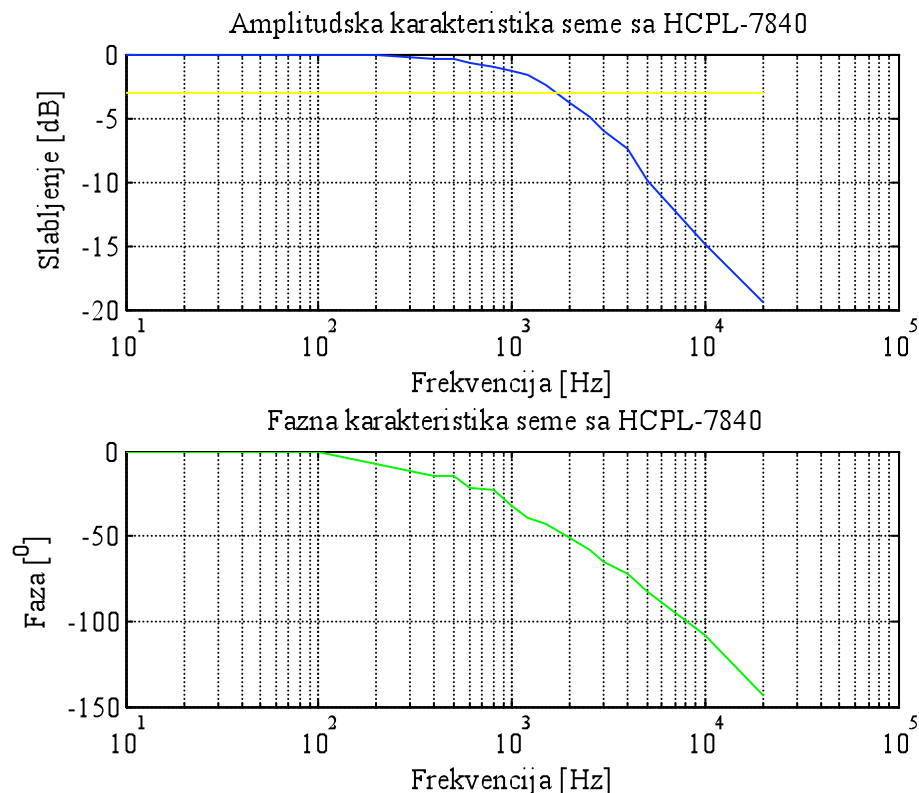
Постигнуте карактеристике реализоване шеме јако су сличне карактеристикама аналогне изолације сигнала у супротном смеру. На следћој слици приказана је функција преноса реализоване изолације.



Слика 4.8 Функција преноса кола за изолацију аналогних сигнала од мотора ка рачунару

Са слике се види висока линеарност, али и засићење које наступа при вредностима улазног напона већим од 6.5V. Уколико тај део занемаримо, и извршимо поређење мерених вредности са вредностима са праве која спаја две најудаљеније тачке на графику, добијамо да је линеарност $21.34 / (3792 - 174) = 0.0059 \approx 0.6\%$. Постигнута линеарност је нешто лошија од линеарности за аналогну изолацију сигнала од рачунара ка мотору, али је свакако задовољавајућа.

Амлитудска и фазна карактеристика приказане су на следећој слици, стим што поново, као и у претходном слушају морамо нагласити да нису мерене истовремено, односно да је амплитудска карактеристика мерена у самом систему, док је фазна мерена у одвојено од посртојења с обзиром на поменути проблем заједничке масе за све сонде осцилоскопа.



Слика 4.9 Фреквенцијске карактеристике реализоване шеме за изолацију аналогних сигнала од рачунара ка мотору

Са слике се види да је -3dB пропусни опсег око 1.6kHz, док је усвојени пропусни опсег у коме амплитуда не пада испод 90% максималне вредности 800Hz.

4.3. Изолација дигиталних сигнала од рачунара ка мотору

Изолација дигиталних сигнала од рачунара ка мотору се уводи као могућност потенцијалне реализације управљања мотором помоћу PWM импулса. Да би се обезбедила могућност софтверски дефинисаног мртвог времена и независног управљања свим прекидачима инвертора, то је реализовано 6 оваквих изолација.

Распоред улазних и излазних пинова овог дела изолације на конекторима прилагодено-изолационе картице приказан је у следећој табели (Табела 4.3 Карактеристике дигиталних изолованих сигнала, од рачунара ка мотору).

Табела 4.3 Карактеристике дигиталних изолованих сигнала, од рачунара ка мотору

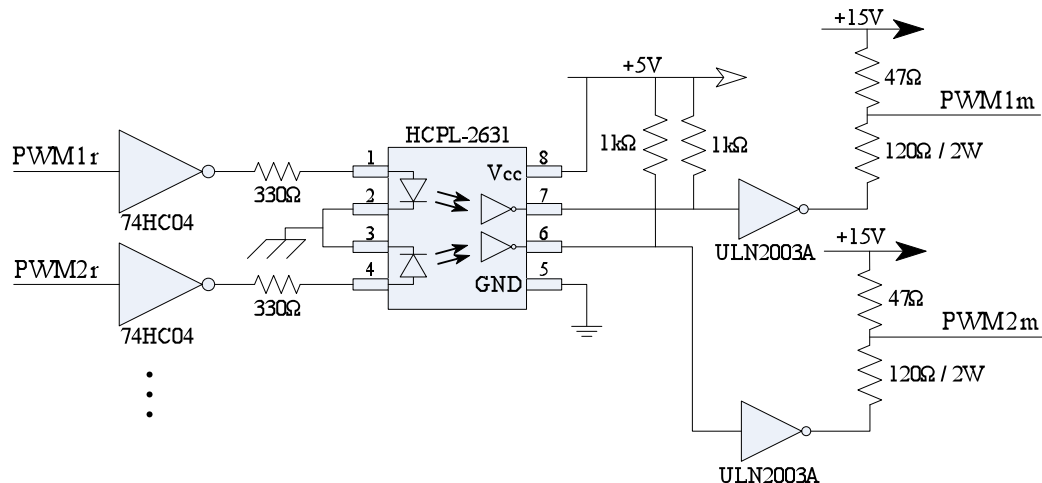
	Страна до рачунара (референтни ниво масе је маса рачунара)		Страна до мотора (референтна маса је K2-GND)	
	Пин JDR1 37 конектора п-к картице тј. J2 конектора п-и картице	Напонски опсег	Пин конектора J1 п-и картице	Напонски опсег
PWM1	Пин 9	0 или 5V	Пин 19	15V или 13.5V
PWM2	Пин 27		Пин 6	
PWM3	Пин 8		Пин 18	
PWM4	Пин 26		Пин 5	
PWM5	Пин 7		Пин 17	
PWM6	Пин 25		Пин 4	

п-к картице – прилагодено-комуникационе картице

п-и картице – прилагодено-изолационе картице

PWM импулсе би задавао рачунар слањем података који би се прослеђивали на излазне пинове самог FPGA чипа. Начин задавања ових импулса се тренутно не специфицира, као и начин шта рачунар шаље до прилагодено-комуникационе картице, једино се захтева да се податак за паљење првог прекидача, PWM1 импулс проследи до 65 пина FPGA чипа, PWM2 импулс до 66 пина FPGA чипа, и тако редом, за свих 6 PWM сигнала, како је то у осталом и специфицирано шемом прилагодено-комуникационе картице. Ово прослеђивање података на одговарајуће пинове FPGA чипа, потребно је обезбедити при реализацији нове функције FPGA чипа и његовом репрограмирању (нову функцију је потребно реализовати у поменутом софтверу *Xilinx Foundation 3.1*, видети ПРИЛОГ В – Функција имплементирана на FPGA чипу о тренутно имплементираној функцији на FPGA чипу).

На следећој слици приказан је део шеме за изолацију дигиталних сигнала од рачунара ка мотору (Слика 4.10). На слици као и на претходнима нису приказани *bypass* кондензатори за стабилизацију напона напајања, који се налазе уз свако напајање, сваког коришћеног чипа. Коришћене вредности су 100nF.



Слика 4.10 Шема изолације дигиталних сигнала, од рачунара ка мотору, коришћењем оптокаплера HCPL-2631

Са шеме се види да струјни капацитет самог пина FPGA чипа није био довољан за укључивање диоде оптокаплера HCPL-2631, па се зато на улазу шеме налази чип инвертора као бафер. Међутим и излаз овог оптокаплера је баферисан драјверским колом са Дарлингтоновим паровима – ULN2003A. Наиме, извод за паљење оптокаплера 4N28 у драјверској картици, могао се обезбедити и директно са излала чипа HCPL-2631, али то није учињено, с обзиром да се овај чип мора напајати са +5V, и да излазни напон мора бити у опсегу од 0 до +7V, што би значило да би напајање од +5V морало давати и потребну струју за паљење оптокаплера у драјверској картици која је реда 80mA. Уколико претпоставимо да се прекидачи користе за задавање PWM-a, то би значило да само половина од њих води, што би у просеку повећало потрошњу извора од +5V за 240mA. Потрошња која би се имала у тој реализацији је око 3 пута већа у односу на тренутну. То би наравно непотребно повећало и загревање напонског регулатора на плочи, који нам обезбеђује ово напајање. На овај начин, убацивањем драјверског кола, тај проблем је превазиђен с обзиром да се потребна струја доводи из напонског извора од +15V који има довољан струјни капацитет за укључивање оптокаплера 4N28 у драјверским картицама за паљење енергетских транзистора. Сам излазни део кола је идентичан излазном транзисторском степену са картице K8 који се, код хистерезисне регулације, користи за укључивање истих тих диода у оптокаплерима на улазима у драјверске картице K3 до K5.

Исти излазни степен изолационог кола се може користити и за паљење горњег, и за паљење доњег транзистора у инверторском мосту, с обзиром да се укључење и искључење врше преко оптокаплера на улазу драјверске картице, чији апсолутни напонски нивои улазне диоде нису међусобно повезани. Због овога је било и могуће реализовати идентичне блокове за паљење свих транзистора у инвертору.

С обзиром да се на прилагодно-комуникационој картици не налазе отпорници који воде од пина којим се задаје PWM, до улаза у реализовану дигиталну изолацију, стање улаза није дефинисано на нулу, како је то очекивани, већ је на високом нивоу. То је при тестирању проузроковало да су у излазном делу кола сви транзистори чипа ULN2003A у засићењу, односно да сви излазни степену воде велику струју (од $\text{око } 15V / (120\Omega + 47\Omega) = 89mA$),

односно да се на излазним отпорницима дисипира снага од око 8W. Како је у дизајну штампе начињена грешка, те се поред блока ових отпорника који се доста греју налазе електролитски кондензатори, то су са плоче одлемљени отпорници од 47Ω , спречено је загревање кондензатора, и на тај начин је продужен њихов радни век.

Иначе, без обзира на овај превид у дизајну, аутор се нада да ће наредни студенти који буду радили на проширењу функција лабораторијске станице "Вектра", успети на прави начин искористе постојећи хардвер и превазиђу поменути проблем.

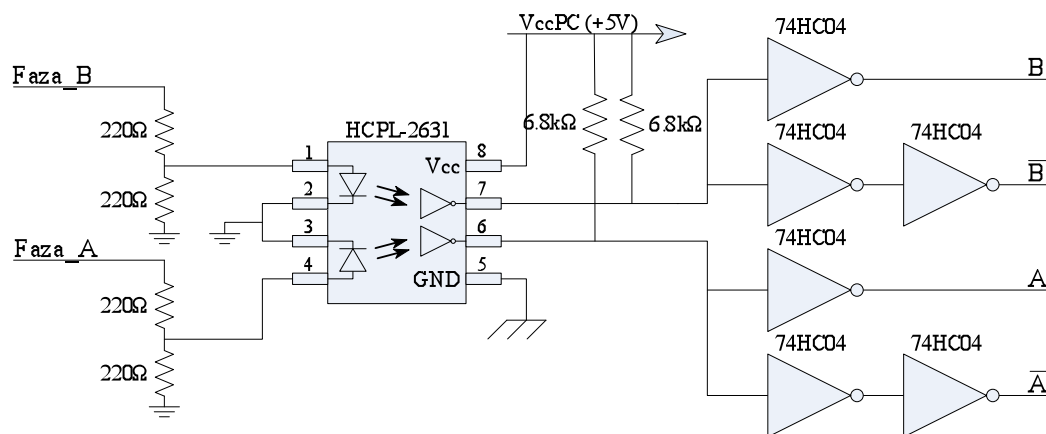
С обзиром на изречено, тестирања овог дела хардвера нису ни вршена.

4.4. Изолација дигиталних сигнала од мотора ка рачунару

Дигитални сигнали који иду од мотора ка рачунару и које треба изоловати су фазе са енкодера. Коришћени енкодер је оптички инкрементални енкодер са 1000 зареза и са два пара сензора. Енкодер је фирме "Zrak" из Сарајева, тип Z 1000 L, напаја се једностраним напајањем од +15V и не препоручује се за веће брзине од 4000 обртаја у минути. Нивои напона фаза енкодера су једнаки нивоу напајања, и износе 0V и +15V. Сходно претходно наведеним подацима максимална учестаност дигиталних сигнала фаза А и В износи: максимална брзина коју може мерити енкодер * број линија по обртају енкодера, односно, $4000 \text{ obrt./min} * 1000 \text{ линија} = 4 \text{ милиона импулса у минути}$, тј. 66.6kHz. Како брзина мотора сигурно неће достићи максималну вредност коју енкодер може мерити, већ је та брзина до 1500 obrt./min, па можемо поставити као услов да изолација дигиталних сигнала треба исправно да ради до учестаности од 25kHz.

Треба још скренути пажњу и на нивое напајања са једне и са друге стране прилагодено-изолационе картице. Напајање доступно са стране прилагодено-комуникационе картице је напајање које се доводи са *game* порта рачунара, тако да се са те стране доводе маса и напајање од +5V. Са друге стране, са лабораторијске станице "Вектра" се доводи симетрично напајање галвански изоловано од напајања рачунара, са напонским нивоима -15V, 0V и +15V, и са референтним нивоом масе дигиталног дела картице K8, које заправо потиче са картице за напајање K2, па је у раду често обележавано са K2-GND. До енкодера се са прилагодено-изолационе картице води посебан *flat* кабл са 4 линије које представљају напајање енкодера (0V и +15V – референтни ниво је K2-GND), а две линије сигнала које се враћају представљају сигнале са фаза енкодера, *Faza A* и *Faza B*.

За изолацију дигиталних сигнала коришћен је оптокаплер HCPL-2631. Реализована шема дата је на следећој слици (Слика 4.11).



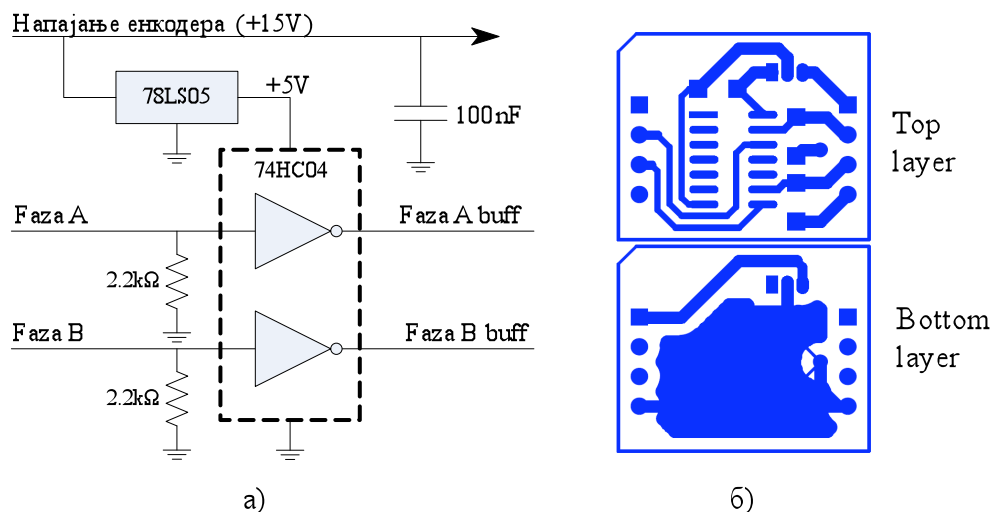
Слика 4.11 Шема за изолацију дигиталних сигнала, са енкодера, од мотора ка рачунару, коришћењем оптокаплера HCPL-2631

На улазу се налази напонски разделник који улазни напон са фаза енкодера, своди на потребан опсег за укључивање диоде оптокаплера. Иако би требало да је овај сигнал 15V, и да је струјни капацитет линија фаза енкодера 8.8mA, с обзиром да се на излазу енкодера налази „НИ“ коло CD4011, чије су карактеристике и наведене, ипак из непознатих разлога није тако. То је проузроковало да фазе енкодера, недовољним струјним капацитетима нису биле у стању да пропусте јаку струју кроз фотодиоду, тако да се излаз оптокаплера није налазио у јаком засићењу. Проблем је разрешен убацивањем, на гране енкодера додатног баферишућег кола, које се састоји само од једног инверторског чипа 74HC04, једног напонског регулатора 78L05, и *bypass* кондензатора. Шема и *layout* овог кола дата је на Слика 4.12. То је повукло промене улазног разделника напона на вредности отпорника приказане на Слика 4.12. Дакле, улази прилагодено-изолационе картице за пријем сигнала са енкодера, намењени су за фазе енкодера чији су напонски нивои 5V, а струјни капацитет око 10mA. Изоловани излазни сигнал се води на инвертор и на двоструки инвертор који представља бафер. На овај начин добијамо изоловану фазу енкодера, али и њен комплемент. Овакав излаз за обе фазе енкодера је непоходан, с обзиром на захтеве прилагодено изолационе картице и коришћени стандард RS-485. С обзиром на велику брзину рада инвертора, у ондосу на сигнал који се преноси кроз њега, можемо рећи да је додатно кашњење које потиче од једног више инвертора у путањи за комплемент фазе, заправо занемарљиво.

Изоловани сигнали фаза енкодера, са прилагодено-изолационе картице, се воде на 25-пински конектор JDR2 прилагодено-комуникационе картице, како је то приказано у следећој табели (Табела 4.4).

Табела 4.4 Везе пинова конектора JDR2 прилагодено-комуникационе картице

Пин	Ознака	Веза са изолованим фазама енкодера
9	A+3	Изолована <i>Faza A</i> енкодера
10	B+3	Изолована <i>Faza B</i> енкодера
22	A-3	Изолована <i>Faza A̅</i> енкодера
23	B-3	Изолована <i>Faza B̅</i> енкодера



Слика 4.12 Баферишуће коло за сигнале фаза енкодера а) шема и б) layout

Реализована дигитална изолација се може користити за пренос дигиталних сигнала високих учестаности. Као релевантан параметар посматрано је кашњење сигнала (*delay*) кроз изолационо коло. У следећој табели приказана је зависност кашњеа од фреквенције дигиталног сигнала.

Табела 4.5 Зависност кашњења изолованог дигиталног сигнала од фреквенције

Фреквенција [kHz]	10	20	30	50	100
Периода [μ s]	100	50	33.3	20	10
Delay [μ s]	0	0.4	0.6	0.7	0.75

Како у нашем систему фреквенција не прелази 25KHz, можемо рећи да реализована структура испуњава тражене захтеве.

4.5. Спецификације конектора прилагодено-изолационе картице

У претходним одељцима описана је функција, и дата је шема сваке од коришћене изолације. Ово поглавље, сада сумира и специфицира све пинове конектора прилагодено-изолационе картице.

Ова картица садржи 3 конектора, и преко два од њих је повезана у "сендвич" са прилагодено-комуникационом картицом, тј. са њеним конекторима JDR1 и JDR2, док трећи конектор представља интерфејс ка погону мотора.

4.5.1. J2 конектор

J2 је 37-пински конектор, који се "огледа" са JDR1 конектором прилагодено-комуникационе картице. Заправо ни један од ова два конектора нема залемљен физички конектор, већ су они међусобно повезани краткоспајајућим жицама.

У следећој табели дате су спецификације пинова овог конектора (Табела 4.6). Како се велики број пинова не користи, то они нису ни приказан у одговарајућој табели.

Табела 4.6 Спецификације пинова конектора J2

Пин	Ознака	Опис	Улаз/ Израз	Напонски опсег
1	Vcc	Напајање прилагодено-изолационе картице са стране до рачунара		+5V
2	GND	Маса прилагодено-изолационе картице са стране до рачунара		0V
7	PWM5	PWM улаз	Улаз	0 или +5V
8	PWM3	PWM улаз	Улаз	0 или +5V
9	PWM1	PWM улаз	Улаз	0 или +5V
12	ADCH2	Иzolована аналогна вредност струје статора или произвољног неког другог посматраног напона референтног нивоа масе K2GND	Израз	0-4.1V
13	ADCH0		Израз	0-4.1V
14	DA1CH0	i_{α}^* - задата референтна вредност за струју статора у $\alpha\beta$ координатном систему	Улаз	0-3.3V
16	DA3CH0	i_{β}^* - задата референтна вредност за струју статора у $\alpha\beta$ координатном систему	Улаз	0-3.3V
25	PWM6	PWM улаз	Улаз	0 или +5V
26	PWM4	PWM улаз	Улаз	0 или +5V
27	PWM2	PWM улаз	Улаз	0 или +5V
30	ADCH3	Иzolована аналогна вредност струје статора или произвољног неког другог посматраног напона референтног нивоа масе K2GND	Израз	0-4.1V
31	ADCH1		Израз	0-4.1V

4.5.2. J3 конектор

J3 је 25-пински конектор, који се ”огледа” са JDR2 конектором прилагодено-комуникационе картице. Као што је већ описано, ни овде се не налази конектор, већ је већ је он повезан краткоспајајућим жицама са конектором JDR2. Овај конектор служи за прослеђивање изолованих сигнала фаза енкодера, у жењеној форми, ка прилагодено-комуникационој картици. Како се са овог конектора користе само 4 пина то ће у следећој табели бити дате само њихове спецификације (Табела 4.7).

Табела 4.7 Спецификације пинова конектора J3

Пин	Ознака	Опис	Напонски опсег
9	Faza A	Иzolована неинвертована фаза енкодера	0 или +5V
10	Faza B	Иzolована неинвертована фаза енкодера	0 или +5V
22	Faza $\overline{A}$	Иzolована инвертована фаза енкодера	0 или +5V
23	Faza $\overline{B}$	Иzolована инвертована фаза енкодера	0 или +5V
25	GND	Маса прилагодено-комуникационе картице	

4.5.3. J1 конектор

J1 је 25-пински конектор који представља интерфејс прилагодено-комуникационе картице са прилагодено-изолационом картицом, ка погону мотора. Путем ове картице прилагодено-изолациона картица добија напајање са стране мотора од +15V, -15V и

референтну масу K2-GND. Преостали сигнали овог порта и њихове карактеристике, приказани су у следећој табели (Табела 4.8).

Табела 4.8 Спецификације пинова конектора J1

Пин	Ознака	Опис	Напонски опсег
1	GND-K2	Маса са погона мотора која долази на п.-и. картицу	0V
2	Faza A	Прва фаза енкодера – директно се доводи са енкодера	0 или +15V
3	GND-K2	Маса са погона мотора која долази на п.-и. картицу	0V
4	PWM6	Сигнали који омогућују PWM регулације покретања мотора. Воде се на улазне оптокаплере 4N28 у драјверским картицама K3, K4 и K5	1,35V/60mA
5	PWM4		
6	PWM2		
7	GND-K2	Маса са погона мотора која долази на п.-и. картицу	0V
8	GND-K2		
9	Ain4	Аналогни улази за сигнале који се обрађују или посматрају на РС-ју. Референтни ниво масе им мора бити K2-GND. Првенствено служе за мерење струја статора.	-8V - +8V
10	Ain2		
11	Vee2	Негативно напајање – -15V у односу на K2-GND	-15V
12	GND-K2	Маса са погона мотора која долази на п.-и. картицу	0V
13	i_{α}^*	Референтна вредност за i_{α}^* струју статора	-10V - +10V
14	Vcc2	Позитивно напајање – +15V у односу на K2-GND	+15V
15	Faza B	Друга фаза енкодера – директно се доводи са енкодера	0 или +15V
16	Vcc2	Позитивно напајање – +15V у односу на K2-GND	+15V
17	PWM5	Сигнали који омогућују PWM регулације покретања мотора. Воде се на улазне оптокаплере 4N28 у драјверским картицама K3, K4 и K5	1,35V/60mA
18	PWM3		
19	PWM1		
20	Vcc2	Позитивно напајање – +15V у односу на K2-GND	+15V
21	Vcc2		
22	Ain3	Аналогни улази за сигнале који се обрађују или посматрају на РС-ју. Референтни ниво масе им мора бити K2-GND. Првенствено служе за мерење струја статора.	-8V - +8V
23	Ain1		
24	Vcc2	Позитивно напајање – +15V у односу на K2-GND	+15V
25	i_{β}^*	Референтна вредност за i_{β}^* струју статора	-10V - +10V

п.-и. Картица – прилагодено-изолациона картица

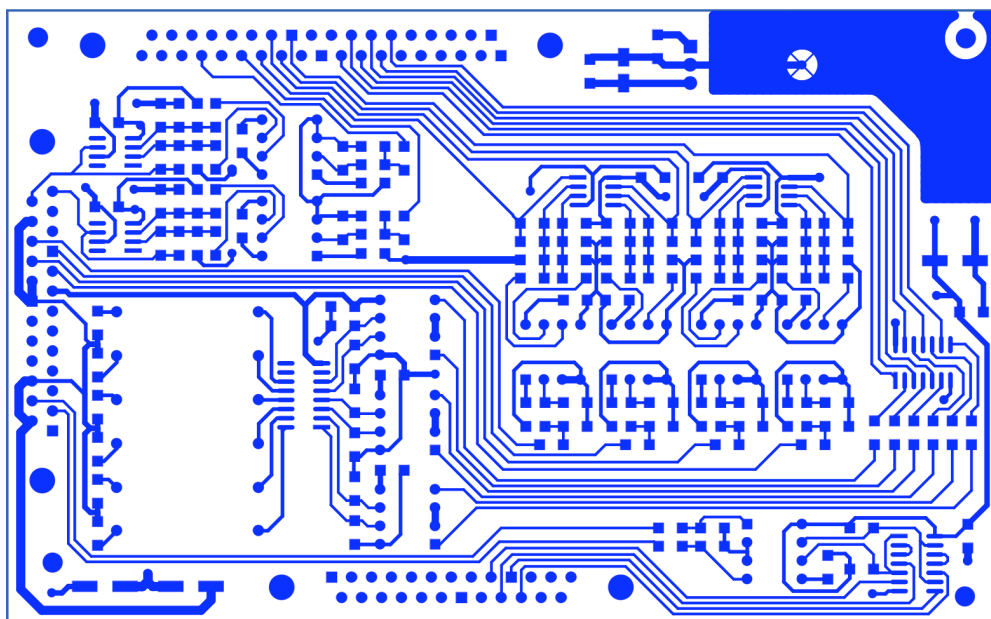
4.6. Layout реализоване прилагодено-изолационе картице

Шема као и сам *Layout* одрађени су у програмском пакету OrCAD 15.7. Шематик је нацртан у *OrCAD Capture*, а сам *Layout* је урађен у *OrCAD Layout Engineer's Edition* програмском делу овог пакета.

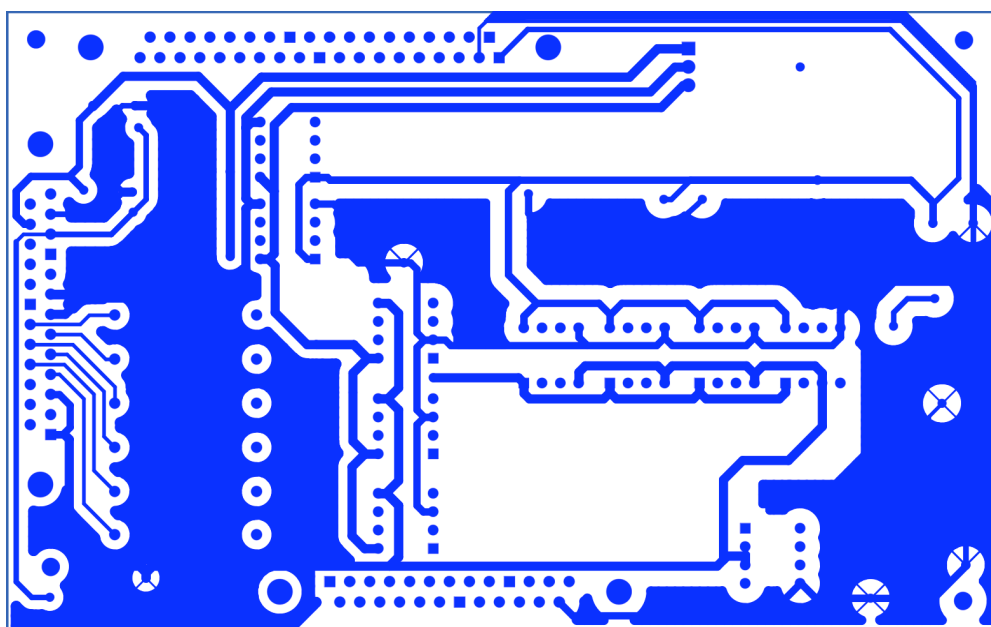
Димензије штампане плоче су 138mm x 85.1mm, и поклапају се са димензијама већ постојеће прилагодено-комуникационе картице. Штампана је двослојна. Минимално растојање између водова, као и минимална ширина вода износе 15mils-а. Завршна дебљина слоја бакра на плочи је 55μm. Дебљина изолатора је 1.5mm, а као диелектрик, коришћен је стандардни изолатор FR4. Сама штампана плоча фабрикована је у Hi Tech Corporation у Македонији.

Приликом израде *Layout*-а водило се рачуна о томе да сигнали различитих нивоа маса, као и саме масе и напајања, чији међусобни напон није стабилан, буду што удаљенији једни од других. То је произвело доста потешкоћа у рутирању *bottom layer*-а у коме се налазе два различита нивоа масе и чак 4 напајања (PC-GND, VccPC, K2-GND, Vcc2, Vee2 и +5V са регулатора).

Изглед *top* и *bottom layer*-а приказан је на следећим сликама (Слика 4.13 и Слика 4.14), а списак коришћених компоненти и њихових *footprint*-ева дат је у прилогу С.



Слика 4.13 *Top layer* ново реализоване прилагодено-изолационе картице



Слика 4.14 *Bottom layer* ново реализоване прилагодено-изолационе картице

5. Начин коришћења „Нове Вектре“

Ово поглавље се састоји од три дела:

- У првом делу описан је ново реализовани код и дате су инструкције за његово коришћење.
- У другом делу описан је начин укључивања и искључивања лабораторијске станице ”Нова Вектра”, као и проблеми који се могу јавити при томе.
- Трећи део описује поступак превезивања каблова и враћање система у првобитни облик, односно онај који био пре израде овог дипломског рада.

Само у овом поглављу, ради лакшег сналажења о којем систему је реч, биће коришћени називи ”Стара Вектра” и ”Нова Вектра” који јасно асоцирају на систем пре и после израде овог дипломског рада.

5.1. Реализовани софтвер и начин његовог коришћења

У овом поглављу је реч о софтверу који се користи за покретање мотора на ”Новој Вектри”. Реч је о коду писаном у програмском језику C, са конзолним интерфејсом. Овде ће бити речи о самом коду и начину његовог коришћења када је систем већ пуштен у рад, а за покретање овог програма и пуштање система у рад погледајте следеће поглавље 5.2. Сам изворни код као и објашњења одговарајућих функција дати су у прилогу D (поглавље 7.4)

Реализовани код се састоји из 3 модула: GLAVNI.C, PORT.C и REGULTR.C, који су обједињени у пројекат INTER.PRJ. Подела на модуле је логичка, тако да се сваком од њих додељена одговарајућа функција. У првом модулу GLAVNI.C, налази функција main у којој се налази ”бесконачна” петља са контролисаним изласком из програма, и у којој су дозвољени прекиди са LPT порта. Овај модул је задужен за комуникацију са корисником, тако да се у њему врши унос података са тастатуре и испис података на конзолни екран. Други модул PORT.C је задужен за пренос података између прилагодено-комуникационе картице и LPT порта, и у потпуности је усклађен са функцијом имплементираном на FPGA чипу, тако да је за тумачење овог дела кода потребно погледати ПРИЛОГ В – Функција имплементирана на FPGA чипу, и посебну пажњу обратити на Временски дијаграми, и то на изглед DATA и STATUS линија LPT порта на њој. Трећи модул REGULTR.C је задужен за

реализацију регулатора и за припрему података који се требају пренети у оквиру периоде пристиглог прекида. Реализовани регулатор је брзински PI регулатор инкременталног типа са пропорционалним дејством измештеним у грану повратне спреге.

Сада ћемо нешто више рећи о првом модулу и функцијама имплементираним у оквиру њега, а потом и о трећем модулу и о параметрима регулатора и променљивама које се користе. О другом модулу неће бити додатног говора с обзиром да је он такав какав је све док се на FPGA чипу налази тренутни код, односно он само врши пренос потребног броја података у одговарајућем редоследу и на онај начин како то захтева функција имплементирана на FPGA чипу.

У оквиру првог модула GLAVNI.C налази се switch структура која у зависности од притиснутог тастера на тастатури обавља различиту функцију. Могуће опције кориснику су:

- ”w” – притиском на тастер w <ENTER> кориснику се даје могућност да унесе нову жељену угаону брзину мотора у обртајима у минути (rpm). Знаком унетог броја је одређен смер обртања мотора. Брзина је софтверски ограничена на $\pm 800\text{rpm}$, тако да ће се за унос брзине веће од 800rpm унета вредност заокружити на 800rpm, док ће се за вредност мању од -800rpm, заокруживање извршити на -800rpm. Након извршеног уноса брзине притиском на тастер <ENTER> мотор ће почети да се врти задатом брзином.
- ”f” – Тастер f је задужен за промену роторског флуksа, односно промену референтне струје статора i_d^* , којој је роторски флуks директно пропорционалан код индиректног векторског управљања (видети поглавље 2 и Слика 2.1). Унета вредност се директно додељује променљивој d у коду која представља дигиталну вредност за i_d^* која се шаље на 12-битни DA конвертор, тако да може узимати вредност од 0 до 4095. Почетна вредност овог параметра је $d=675$ и представља номиналну вредност за ову струју. Параметри за номиналне вредности за струје i_d^* и i_q^* су преузети из упутства за ”Стару Вектру” [1] и износе: $i_{dn}=1.93\text{A}$ и $i_{qn}=2.24\text{A}$. Коришћене вредности у коду за d и q које представљају дигиталне еквиваленте ових бројева добијене једноставним експериментом. Вршена је промена параметара d и q уз задржавање константног односа $d/q = i_{dn}/i_{qn}$ док мерена струја у фази статора није достигла номиналну вредност од 2.1A. При томе се водило рачуна да брзина буде што ближа номиналној тако да су параметри одређени при брзини од око 800rpm. Добијени параметри за номиналне вредности за d и q су $d=675$ и $q=783.42$.
- ”p” – Притиском на тастер p може се унети нова вредност за константу пропорционалног појачања КР. Почетна вредност овог параметра подешена је на експериментално одређену вредност при којој се добија стриктно апериодичан одзив брзине на step побуду за одговарајућу периоду регулације брзине која је дефинисана променљивом NN у REGULTR.C модулу. Приликом уноса нове вредности овог параметра не треба нагло одступати од претходне вредности која се може прочитати са екрана.

- "i" – Притиском на тастер *i* може се унети нова вредност за константу интегралног појачања KI. Почетна вредност овог параметра подешена је на експериментално одређену вредност при којој се добија стриктно апериодичан одзив брзине на *step* побуду за одговарајућу периоду регулације брзине која је дефинисана променљивом NN у REGULTR.C модулу. Приликом уноса нове вредности овог параметра не треба нагло одступати од претходне вредности која се може прочитати са екрана.
- "e" – Притиском на тастер *e* врши се постављање струја статора на нулу, и контролисани излазак из програма.

У оквиру трећег модула REGULTR.C врши се припрема података за пренос, на основу података добијених из претходно извршеног преноса. Периода прекида $T_{\text{INTERRUPT}}$ се може софтверски задавати помоћу променљиве *perioda*, која је тренутно постављена на вредност 1000, што дефинише периоду прекида од 290 μ s (мерено на осцилоскопу). Постављена периода нешто је већа од минималне хардвером дефинисане вредност од 204 μ s (видети поглавље 7.2.15) која је потребна FPGA чипу да изврши потребан пренос података, и AD и DA конверторима да изврше потребне конверзије. При тестовима, са повећавањем параметра *perioda*, повећавала се и бука мотора, тако да је усвојена ова условно минимална вредност тог параметра *perioda*=1000.

У оквиру овог модула дефинисан је и параметар NN који служи за повећање прецизности одређивања мерене брзине, и који дефинише периоду брзинског регулатора. Тренутна вредност овог параметра у коду је NN=60, тако да је периода брзинског регулатора 60x290 μ s=17.4ms.

У оквиру овог модула имплементиран је и PI регулатор, а његови параметри одређени су експериментално. Оптималност ових параметара се покушала подесити према упутству за лабораторијску вежбу "Испитивање динамичких својстава дигитално регулисаног брзинског сервомеханизма путем рачунарских симулација" [4], али се испоставило да се они нису у потпуности одговарајући па се приступило експерименталној методи пробања. У тестовима брзина ротора је мењана са +500rpm на -500rpm, и обрнуто, а као излаз су на осцилоскопу посматране вредности за момент и брзину које се у овом делу кода путем променљивих *dac3* и *dac4* шаљу на DA конверторе тако да се конкретно ове вредности могу посматрати на пиновима 32 и 34 37-пинског конектора JDR1 прилагодно-комуникационе картице. Као критеријум оптималности узете су вредности када је одзив регулисане брзине стриктно апериодичан, када је бука мотора мања, односно када су осцилације момента у стационарном стању мале, и када нема осцилација око жељене вредности у равнотежном стању. Извршено је неколико мерења у зависности од периоде регулације и она су приказана у следећој табели (Табела 5.1). Мерења су приближна, а освенчено поље у табели може се сматрати за оптималне параметре при одговарајућој периоди брзинског регулатора.

Табела 5.1 Експериментално одређене оптималне вредности за KP и KI у зависности од периоде регулације брзине

NN=20			NN=40			NN=60		
KP	KI	квалитет	KP	KI	квалитет	KP	KI	квалитет
>48	4	јака бука	42.4	4	бучно	42	6	бучно
>48	3	бучно	36	3	ок	34	4	ок
>40	2	ок	32.5	2.5	ок	31.5	3	оптимум
32	1	оптимум	30	2	оптимум	24	2	ок
18	0.5	осц.	17	1	осц.	17	1	осц.

У оквиру модула REGULTR.C врши се и софтверска контрола мерење брзине која треба да спречи нежељене последице које могу наступити погрешним уносом неког од параметара који би проузроковао превелику брзину мотора, тако да при мереној брзини већој од 1200rpm, долази до активације софтверске заштите и постављања струје мотора на нулу, и изласка из програма.

5.2. Процедура укључења и искључења ”Нове Вектре”, проблеми који се јављају при томе и решења тих проблема

Када је хардвер повезан у конфигурацију ”Нове Вектре” укључење се врши према следећој процедури:

- 1) Укључи се напајање за радни сто.
- 2) Укључи се рачунар и подигне оперативни систем Windows98®.
- 3) Укључи се напајање управљачког кола инвертора.
- 4) Покрене се програм INTER.EXE који се налази у директоријуму C:\Bc20\OUT уколико се жели само пуштање мотора у рад без посматрања кода. Уколико се жели и посматрање и едитовање кода потребно је отворити компајлер BC.EXE који се налази у директоријуму C:\Bc20\BIN и из њега отворити пројекат INTER.PRJ који се налази у директоријуму C:\Bc20\Sources\Inter20. Затим је потребно извршити компајлирање кода Compile->Build All, покретање кода Run->Run.
- 5) Да би се уверили да ли комуникација исправно ради (енергетски део још увек није укључен) потребно је завртети осовину мотора и видети да ли има промене на екрану вредности i_q која се исписује, или уколико на осцилоскопу посматрамо задаване струје i_α^* или i_β^* у том случају генерисаће се одговарајуће синусоиде на осцилоскопу. Уколико до овога не дође десио се ПРОБЛЕМ 1.
- 6) Уколико смо успешно завршили претходну тачку, што казује да комуникација ради, односно да рачунар исправно прихвата прекиде, угасимо програм притиском на тастер ”e”, затим угасимо сигурносни прекидач који гаси десну половину стола, а затим укључимо и напајање енергетског кола инвертора. Потом укључимо

сигурносни прекидач за исту половину радног стола. Након 15-ак секунди преспојимо отпорник кроз који су се напунили кондензатори у енергетском колу инвертора.

- 7) Поново покренемо програм као у тачки 4).
- 8) Управљамо програмом на начин описан у претходном одељку 5.1.

Процес искључења је обрнут, односно:

- 1) Притисне се тастер "е" за контролисани излазак из програма.
- 2) Искључи се сигурносни прекидач за напајање десне половине радног стола.
- 3) Искључи се напајање енергетског кола инвертора, и отвори се прекидач који је преспајао отпорник кроз који су се напунили кондензатори у енергетском делу кола инвертора.
- 4) Затворе се сви програми и искључи се рачунар.

Објашњење ПРОБЛЕМА 1. Проблем 1 је проблем који се јавља услед тога штоFPGA чип на прилагодено-комуникационој картици поседује подразумевану периоду интерапта који се јавља чим картица добије напајање. Како она добија напајање при самом укључењу рачунара, то и прилагодено-комуникациона картица тражи прекид од самог почетка, пре него што се и сам оперативни систем подигао. Ово условно речено "збуни" рачунар, тако да комуникација описана у процедури укључивања лабораторијске станице "Нова Вектра" највероватније неће прорадити из прве.

РЕШЕЊЕ: Једно од могућих решења овог проблема је да се напајање извади из конектора на прилагодено-комуникационој картици пре укључивања рачунара, тј. пре тачке 1), а затим да се поново врати у току тачке 5). На овај начин заправо вршимо ресетовање FPGA чипа. Друго решење је слично претходном, а то је да се у току тачке 5) напајање и извади и врати. Уколико ово не успе процедуру можемо поновити.

Срећна околност је што када једном проради комуникација, исправно ће радити за све време док је рачунар укључен.

Други проблем који се може јавити у току рада је да дође до нелегалног завршетка програма. У том случају при следећем покушају нећемо бити у могућности да отворимо пројекат. Решење овог проблема је јакo једноставно и састоји се у брисању фолдера INTER20 на локацији C:\Bc20\Sources\INTER20, и деархивирању .zip фајла INTER20-VektraNew.zip у фолдер са старим именом C:\Bc20\Sources\INTER20.

5.3. Повезивање система у конфигурацију ”Нова Вектра” и враћање у конфигурацију ”Стара Вектра”

Уколико се систем жели повезати у конфигурацију ”Нова Вектра” потребно је у одговарајући рачунар са одговарајућим софтвером, оперативним системом Windows98®, LPT и GAME портом, на LPT порт директно повезати сендвич картицу, док је са 1, 8, 9 или 15 пина GAME порта рачунара потребно довести напајање на 1. пин J2 конектора прилагоднo-комуникационе картице.

На други слободан крај ”сендвич” картице, односно на 25-пински конектор прилагоднo-изолационе картице, потребно је повезати кабл који излази из картице K9 струјно контролисаног напонског инвертора, и који има одговарајући конектор. Из овог конектора наставља се 4-жилни *flat* кабл са женским конектором, као за серијски порт, који треба повезати на енкодер.

Ово би била процедура повезивања система у конфигурацију ”Нова Вектра”.

Уколико се жели систем вратити у конфигурацију ”Стара Вектра” потребно је откачити два поменута конектора из енкодера и прилагоднo-изолационе картице, пронаћи други кабл који излази из картице K9 струјно контролисаног напонског инвертора, и на њему уочити 31-пински конектор који треба повезати на одговарајуће место на прилагоднoј картици 2 (једно једино са тим бројем пинова), а 4-жилни *flat* кабл који наставља из повезаног конектора и који се завршава са женским конектором, као за серијски порт, треба повезати на енкодер.

Након овога систем је повезан у конфигурацију ”Стара Вектра”.

6. ЗАКЉУЧАК

Реализована прилагодно комуникациона картица представља велико унапређење лабораторијске станице "Вектра" и даје јој читав низ нових могућности управљања. Снага заправо лежи у прилагодно-комуникационој картици и њеном FPGA чипу, али реализована изолациона картица свакако унапређује и проширује њене могућности. Добијена "сендвич" картица, се може користити у потпуности или својим делом и у другим апликацијама. Софтверским модификацијама самога кода и конфигурације FPGA чипа она постаје универзални интерфејс РС рачунара, који помоћу његовог LPT порта комуницира са произвољним уређајем. Флексибилност се условно, уколико је то потребно, може повећати променама напонских опсега интерфејса ка уређају, једноставном заменом одговарајућих отпорника на прилагодно-изолационој картици.

Реализована картица допушта да будућа унапређења система буду сада само софтверског типа. Нека примарна унапређења, би која би се тичала тренутног програма, су побољшање интерфејса, отварање могућности за приказ резултата на екрану, уместо на осцилоскопу. Такође битна ствар је и отклањање проблема описаног у поглављу 5.2, а који се мора решити репрограмирањем FPGA чипа.

Један од будућих задатака је и омогућавање да се написани код може користити и под Windows-ом XP, с обзиром да садашња реализација ради под оперативним системом Windows 98.

Свеукупно можемо рећи да је реализована картица побољшала тренутну реализацију, у смислу осавремењивања хардвера и софтверског окружења, али и отворила читав низ могућности за нове реализације управљања и аквизиције параметара из погона мотора. Такође, првобитно постављени циљ, управљања асинхроним мотором помоћу РС рачунара, у овом раду је остварен!

7. Прилози

7.1. ПРИЛОГ А - Паралелни (LPT) порт РС рачунара

Паралелни интерфејс – LPT порт, представља стандардни интерфејс већине РС рачунара. Типично су подржана до 3 паралелна интерфејса, који се означавају са LPT1, LPT2 и LPT3. Првобитна намена била му је повезивање РС рачунара са линијским штампачем, па отуда и ознака порта LPT (од *Line Printer Terminal*). Овај интерфејс производ је IBM-а, а штампачи за које је прављен интерфејс били су фирме *Centronics* (која више не постоји), тако да је и сам назив интерфејса првобитно био *centronics* интерфејс. По оригиналној концепцији и реализацији пренос података преко овог интерфејса био је могућ само у једном смеру од рачунара ка периферији – штампачу. Данас срећом то више није тако, па је опсег примена овог порта далеко шири.

Код паралелног интерфејса дигитални податак дужине n бита може се пренети у току једног сигнала такта путем n линија за пренос. Конкретно, код LPT порта број линија за пренос података износи 8, тако да је омогућен једновремени пренос податка величине једног бајта. Поред линија за податке, паралелни интерфејс садржи и одређени број управљачких линија (контролних и статусних). Контролне линије порта сигнализирају пријемнику да су подаци важећи, док статусне враћају повратну информацију предајнику – рачунару, да су подаци примљени. На тај начин се врши синхронизација између рачунара и уређаја прикљученог на паралелни порт.

Овај вид комуникације, неког периферног уређаја и рачунара, захтева да сама периферија садржи микропроцесор. Микропроцесор, тј. интелигентни уређај, који је на коришћеној прилагодено-комуникационој картици реализован као FPGA (*Field-Programmable Gate Array*) чип, у оваквим системима поред тога што управља комуникацијом, преузима и део посла који се односи на обраду података, и на тај начин делимично растеређује сам рачунар.

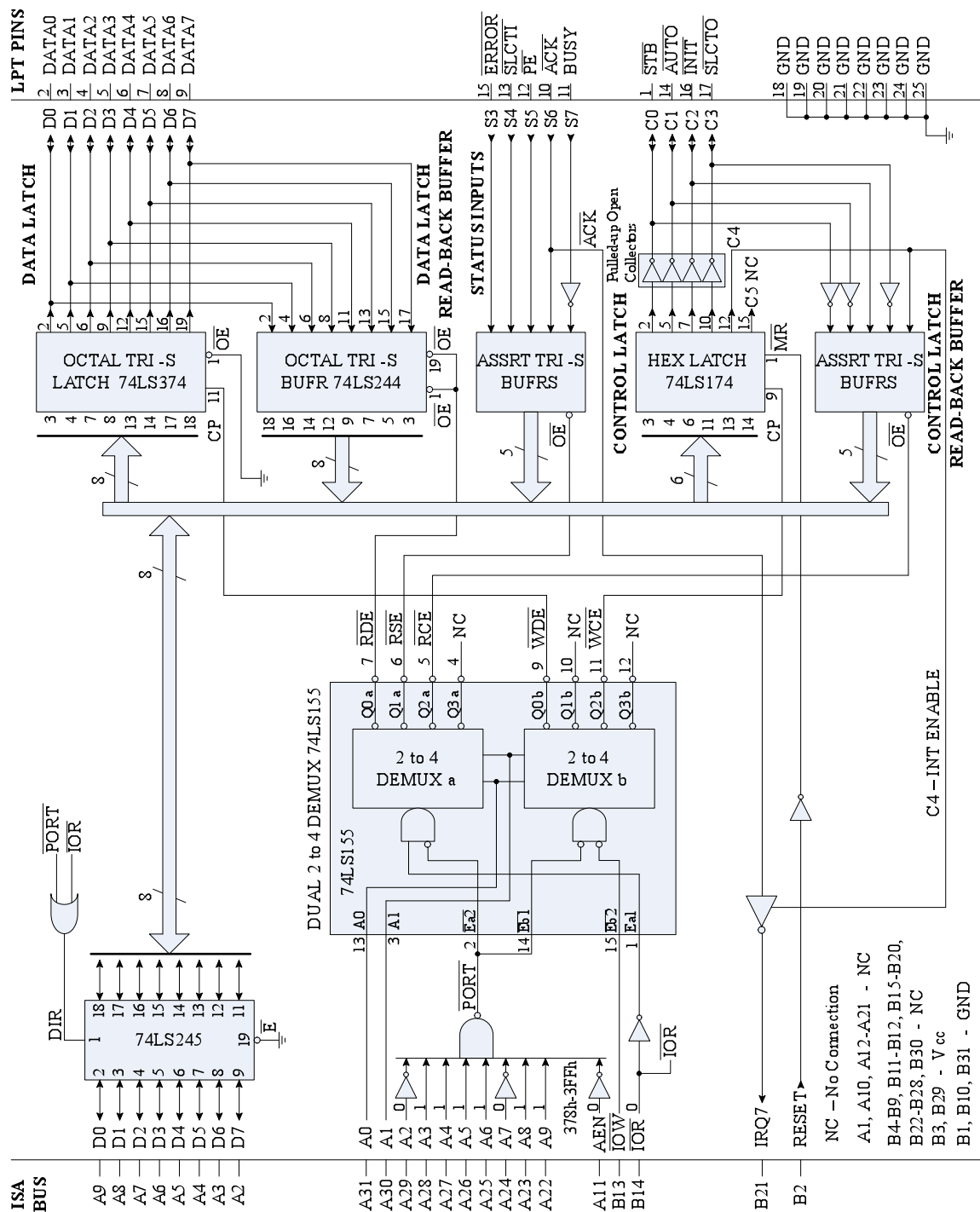
Коришћење првобитног паралелног интерфејса за пријем података је, због неуниформности хардверског решења интерфејса, било нешто компликованије. Пренос података био је предвиђен само у једном смеру, од рачунара ка штампачу. Сама реализација интерфејса је била таква да није остављена могућност његове једноставне модификације у бидирекциони. Наиме, слање података са паралелног порта рачунара врши се преко

тростатичког транспарентног леча 74LS374 (Слика 7.1). Контролни улаз $\overline{OE}$ (*Output Enable*) овог леча је хардверски дефинисан, тј. физички је повезан на масу па су излази из леча стално активни, чак и у периодима када се не шаљу подаци из рачунара. Тако да, иако постоји хардвер који би могао прихватити податке (тростатички леч 74LS244), линије за пренос података увек су окупиране од стране леча 74LS374, и понашају се као излазне. Уколико контролни улаз $\overline{OE}$ леча 74LS374 не би био физички повезан на масу, већ на излаз неког од регистара, овај сигнал би се могао софтверски контролисати, контролом одговарајућег бита у том регистру. Сетовање коришћеног бита, поставило би сигнал $\overline{OE}$ на логичку јединицу, што би резултовало постављањем излаза леча у стање високе импедансе. Тада би, с обзиром да је хардверски већ подржан, био могућ и пријем података. Дакле, тиме би постигли двосмерни пренос података, са софтверском контролом смера преноса, путем 8 линија за пренос података паралелног порта. Управо ова модификација је и извршена код наредних верзија паралелног порта.

У самом раду није коришћена ова модификација, која је могућа али непрактична, већ је бидирекциони пренос остварен коришћењем статусних линија.

Код рачунара серије PS/2, IBM је увео бидирекциони паралелни интерфејс, управо описаним начином модификације, где се смером преноса података управља посебним битом (*Direction bit*, 5 бит контролног регистра на Слика 7.1, пин 15 74LS174 леча). Међутим, и поред очигледне предности коју је ново решење донело, стандардни првобитни интерфејс је и даље јако често заступљен. У намери да обезбеди што ширу примену паралелног интерфејса, асоцијација IEEE је донела стандард IEEE-P1284 којим се дефинише 5 могућих начина рада овог интерфејса: 1) Компатибилни рад (код оригиналних паралелних интерфејса), 2) Нибл рад (коришћењем статусних улаза), 3) Бајт рад (бидирекциони пренос путем линија за податке), 4) ECP (*Extended Capabilities Port*) – паралелни интерфејс проширених могућности (могућ је и асинхрони бидирекциони пренос), 5) EPP (*Enhanced Parallel Port*) – усавршени паралелни интерфејс (преко 8-битне паралелне магистрале преносе се и адресе и подаци).

Код већине савремених РС рачунара омогућено је конфигурисање паралелног интерфејса за потпуни бидирекциони рад али се исто тако може срести и првобитна верзија интерфејса. Паралелни интерфејс који задовољава IEEE-P1284 ће се по укључењу напајања наћи у компатибилном моду рада. Слањем одговарајућих команди и читавањем одговарајућих статуса, рачунар открива какав је тип интерфејса саме периферије. Програм који управља паралелним интерфејсом у рачунару треба да, након што испита којим типом интерфејса располаже периферни уређај, дефинише одговарајући начин рада паралелног интерфејса [3].



Слика 7.1 Електрична шема LPT1 порта

Стандардом (IEEE-P1284) остављена је могућност повезивања рачунара или периферија које имају интерфејс реализован по првобитном – *centronics* стандарду, али они могу користити само прва два начина рада, то јест компатибилни или нибл рад. Међутим пренос података не мора бити ни према једном од стандардном дефинисаних модова, већ се може целокупни порт посматрати само као скуп регистара у које се нешто уписује, или из

којих се нешто чита. Управо један такав начин преноса коришћен је у самом раду. Сваки, стандардом дефинисани мод преноса, предвиђа одређену синхронизацију преноса и проверу да ли су подаци коректно пренети, док у методу коришћеном у раду нема никакве сигнализације и провере. Исправност преноса података заснива се на добро процењеном тајмингу интерфејса самога порта и саме периферије. Односно уколико, на пример, шаљемо податке од рачунара ка периферији, пре слања наредног податка нећемо вршити проверу да ли је претходни прихваћен, већ ћемо тај наредни податак послати након одговарајућег времена за које смо сигурни да је периферија успела да прихвати и обради претходни податак. Такође и сигурност преноса података од периферије ка рачунару се врши на исти начин, односно без додатне синхронизације већ само на основу процене брзине рада самих уређаја. Такав вид комуникације је намерно изабран, са циљем да пренос података буде што бржи.

Статусне линије, које обавештавају рачунар о статусу штампача, су улазне линије у порт, па како наша периферија није штампач, већ прилагодено-комуникациона картица, те „статусне“ линије су искоришћене као улазне линије у рачунар за пријем података са прилагодено-комуникационе картице. На тај начин постигнут је бидирекциони пренос, и избегнут је могући проблем да се на коришћеном рачунару налази порт старије генерације, а да сама периферија захтева комуникацију по неком од напреднијих начина рада паралелног интерфејса.

Сада комплетности ради, а и из разлога што је сам интерфејс прилагодено-комуникационе картице реализован слично интерфејсу штампача у следећој табели (Табела 7.1 Називи и функције линија паралелног интерфејса РС рачунара) дати су називи и функције линија првобитног паралелног интерфејса.

Табела 7.1 Називи и функције линија паралелног интерфејса РС рачунара

Група	Назив	Општи назив	Смер	Функција код првобитног LPT порта
Изразне линије	D0-D7	D0-D7	Од рачунара ка штампачу (или D0-D7 бидирекциони)	Линије за пренос података; користи се позитивна логика.
	$\overline{\text{STROBE}}$	C0		Строб импулс, активна силазна ивица. Типично време трајања импулса је 1 μs . Користи се за упис податка у штампач.
Управљачке линије	$\overline{\text{AUTOFEED}}$	C1		Низак ниво изазива, након знака <i>carriage return</i> , прелазак на нови ред (није потребан <i>line feed</i> карактер).
	$\overline{\text{INIT}}$	C2		Низак ниво ресетује принтер и брише бафер штампача.
	$\overline{\text{SELIN}}$	C3		Низак ниво омогућава штампање.
	$\overline{\text{ACK}}$	S6	Од штампача ка рачунару	Импулс потврде, активна силазна ивица. Типично време трајања импулса је 5 μs . Генерише га штампач када прихвати податак.
Статусне линије	BUSY	S7		Висок ниво када штампач није спреман да прими податак.
	PE	S5		<i>Paper End</i> – висок ниво када штампач нема папира.
	SEL	S4		Висок ниво када је селектовани штампач спреман.
	$\overline{\text{ERROR}}$	S3		Низак ниво ако постоји грешка у раду штампача.

7.2. ПРИЛОГ В – Функција имплементирана на FPGA чипу

7.2.1. Увод

Овај део текста покушава да читаоцу приближи и на високом нивоу опише функцију коју врши FPGA чип. Текст је направљен према *Xilinx* пројекту, који садржи VHDL и шематик фајлове. Сам пројекат је написан у *Xilinx Foundation 3.1* верзији софтвера, што може представљати проблем при измени ових фајлова и репрограмирању чипа, с обзиром да новије верзије истог софтвера више не садрже податке за коришћени *Spartan* чип, у својим библиотекама (актуелна верзија је *Xilinx Foundation 9.2* – 2008. година).

Поменути пројекат се налази у додатним електронским материјалима, у фолдеру E\Xilinx\lpt1_on_S20xilinx.zip, и за његов поновни преглед, и евентуалне измене потребно је: инсталирати инсталирати *Xilinx Foundation 3.1*, и из њега деархивирати поменути .zip file. Упутство за инсталацију поменутог софтвера налази се на приложеном диску.

Како је поновни преглед овог пројекта прилично компликован, овим текстом се желело олакшати читаоцу да се упозна са функцијом коју врши реализована структура на FPGA чипу, а да при томе читалац не мора изучавати коришћени софтвер, нити познавати програмски језик VHDL.

На почетку морамо назначити разлику између две коришћене врсте *file*-ова (шематик и VHDL), и описати структуру самог пројекта. Циљ коришћеног (*Xilinx Foundation 3.1*) и сличних софтверских пакета, је да софтверски (VHDL *file*-ови) и ”хардверски” (*schematic file*-ови) описану функцију, имплементирају на одговарајућем хардверу програмабилног (FPGA) чипа. Хијерархијски највиши *file* у пројекту назива се *top-level file*, и у овом пројекту он је реализован као *schematic file*. Треба напоменути да новије верзије софтвера, у потпуности искључују коришћење *schematic file*-ова, већ је једини стил описа жељене структуре VHDL код.

7.2.2. Top-level schematic file

На следећој слици (Слика 7.2), приказана је шема реализоване структуре на највишем нивоу, односно *Top-level schematic*.

На слици се види да су улази и излази из система, сами пинови FPGA чипа (обележени масним црним ознакама **Pi**). Ради лакшег сналажења поред одређених пинова на слици се налазе и називи сигнала са шеме којој овај FPGA чип припада (обележени плавим словима). И наравно, свака слика је у одговарајућем делу искоментарисана словима бордо боје³.

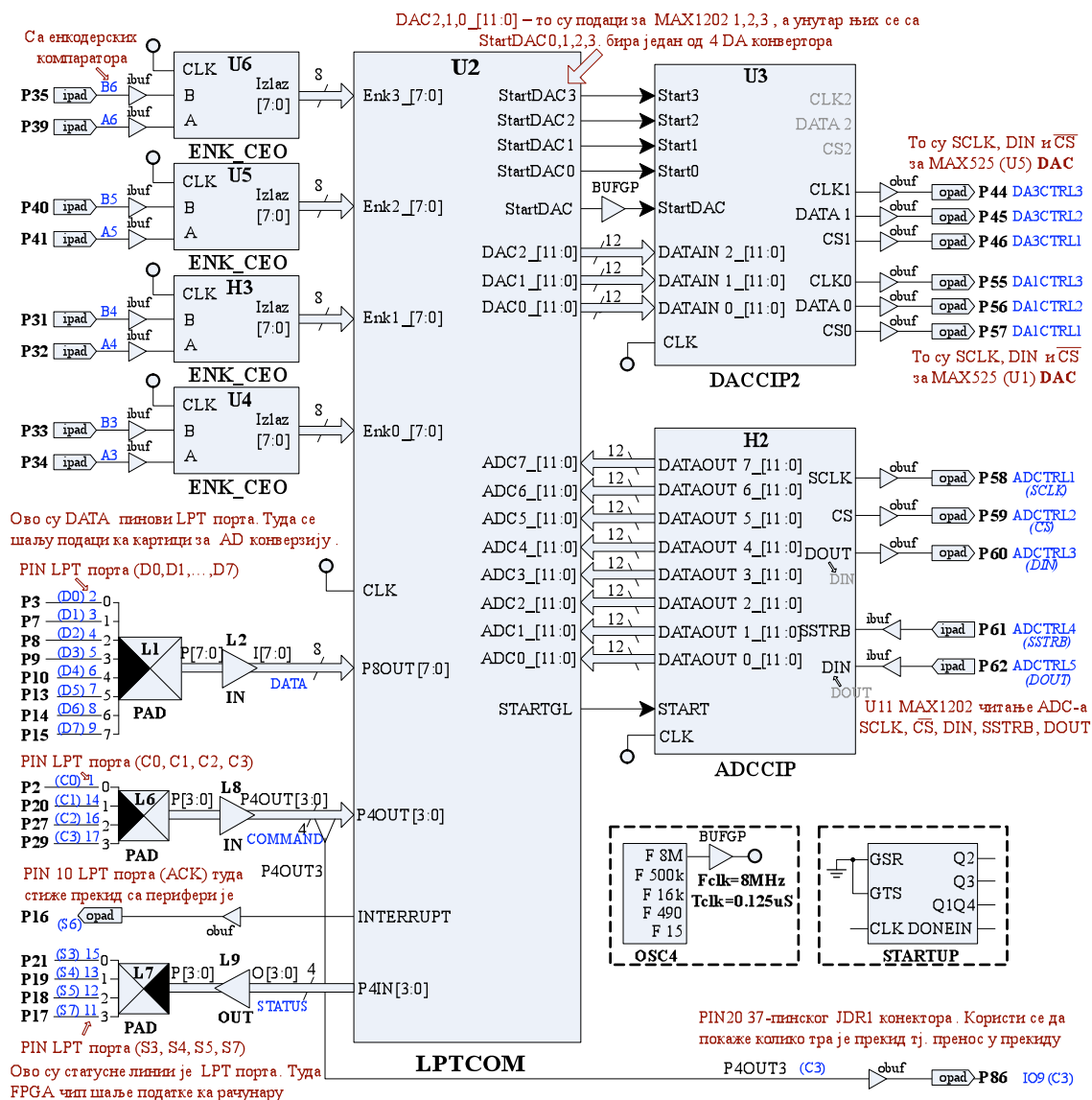
Називи блокова и сигнала одговарају оригиналним називима из самог пројекта.

На слици се једноставно уочава интерфејс ка паралелном порту рачунара, ка пријемницима енкодерских сигнала, и ка DA и AD конверторима. Између централног блока LPTCOM и поменутих интерфејса, постоје одговарајући блокови који врше прилагођење сигнала самом интерфејсу (бафери, ENK_CEO, DACCIP2, ADCCIP). (За потпуно разумевање

³ Ова правила важиће за све слике у овом поглављу

потребно је погледати одговарајућа поглавља о LPT порту, и компонентама SN65LBC173, MAX525, MAX1202 (3.1, 3.3.5, 3.3.3, и 3.3.4), и изучити њихове интерфејсе.) Поред поменутих блокова, на слици се уочавају и блокови за STARTUP, који врши дефинисање почетног стања регистра унутар чипа при укључењу (на овај начин, почетно стање свих регистра постављено је на нулу), и блок осцилатора OSC4, који дефинише локални такт чипа, са којег видимо да се користи такт на 8MHz, и да је то радна учестаност чипа.

Функције преосталих блокова (LPTCOM, DACCIP2, ADCCIP, ENK_CEO) биће детаљно изанализиране у наставку текста.



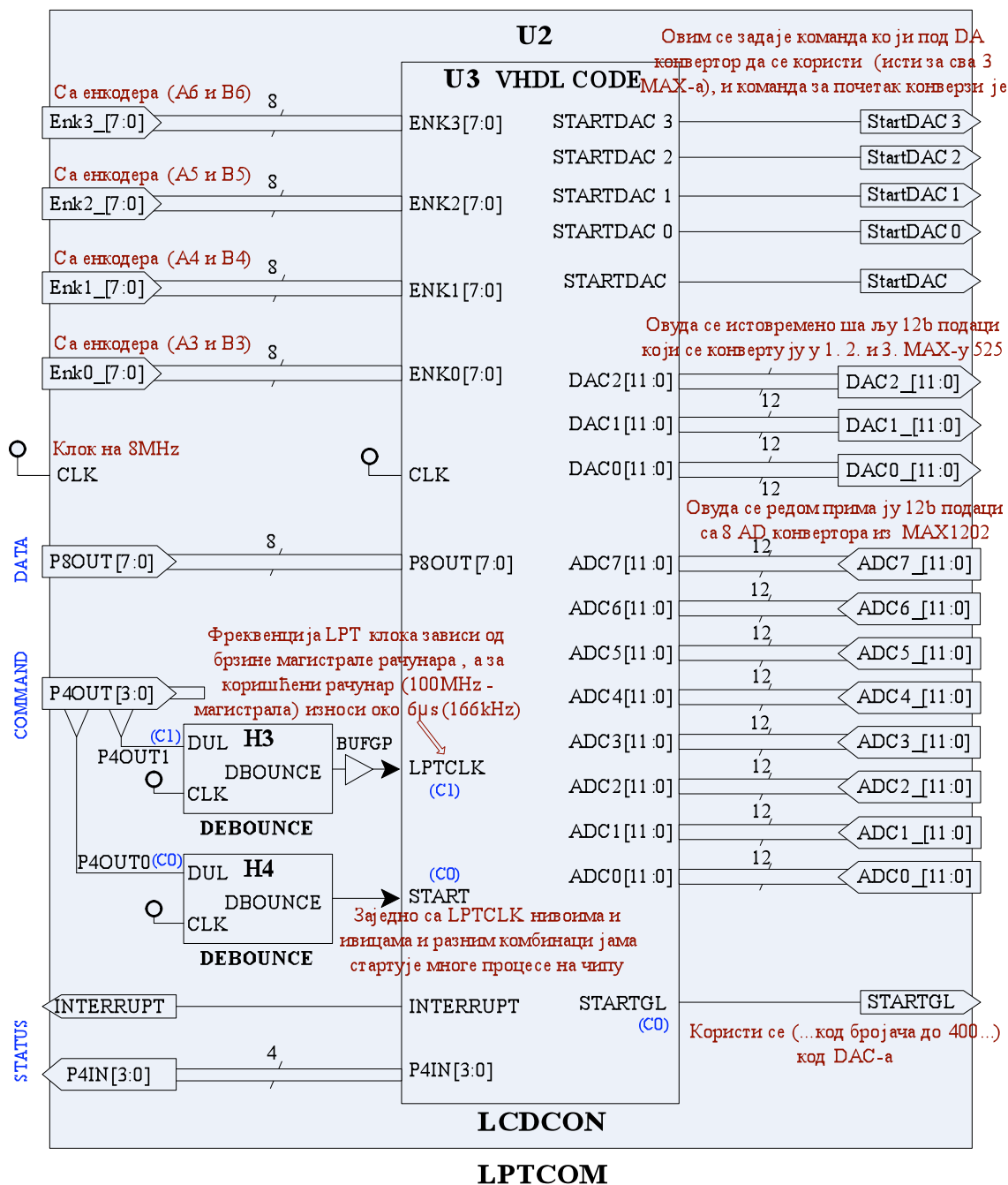
Слика 7.2 Top-level file FPGA чипа

ВАЖНА НАПОМЕНА! На крају овог поглавља се налази слика временских дијаграма готово свих сигнала са овог чипа, тако да се саветује да се при читању овог текста, анализи слика, коментара, и тумачењу одговарајућих сигнала ова слика увек има при руци, и да се

одговор често потражи и на њој. На поменутој слици су одговарајућим блоковима (са леве стране) одвојени сигнали према фајловима из којих потичу. Да би читаоца подсетили да сигнале посматра на поменутој слици, у правом тренутку, поред сваког поднаслова чији су сигнали приказани на дијаграму временских сигнала, стављен је одговарајући знак ().

7.2.3. LPTCOM

Шема LPTCOM блока дата је на следећој слици (Слика 7.3). Ово је блок који је описан *schematic file*-ом, тј. шемом унутар које се налазе нови одговарајући блокови. Ти блокови су DEBOUNCE (описан *schematic file*-ом) и LCDCON (описан *VHDL file*-ом), а њихове функције ће бити накнадно објашњене.

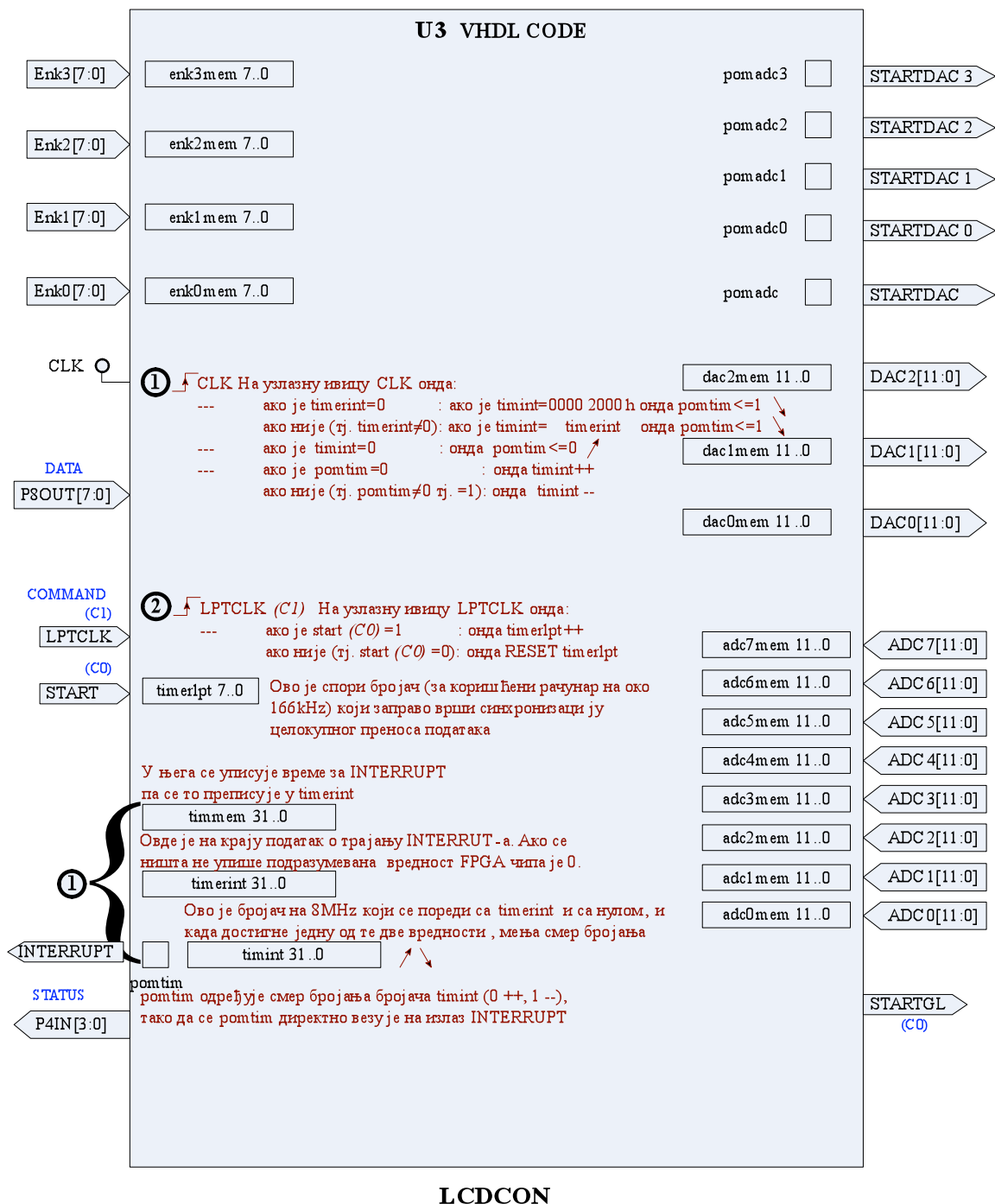


Слика 7.3 LPTCOM шематски приказ

7.2.4. LCDCON

LCDCON ⁿ

LCDCON блок је описан VHDL кодом па ће његово објашњење овде бити приказано као сликовити опис кода. Како се VHDL код за опис овог блока састоји из доста процеса, те како коментари не би били збијени на једној слици, то ћемо опис свих процеса разложити на две слике са одговарајућим коментарима за сваки од њих (Слика 7.4 и Слика 7.5).



Слика 7.4 Анализа 1. и 2. процеса из VHDL кода за LCDCON

1. Процес 1 служи за генерисање периодичног сигнала одговарајуће периоде која представља периоду задавања прекида рачунару од стране прилагодено-комуникационе картице. Податак о трајању овог интервала се може задати програмски из .С кода, и тај податак ће бити смештен у *timerint* променљиву. Уколико се тај податак не преноси односно не задаје на поменути начин, вредност *timerint* променљиве неће бити недефинисана, већ ће бити =0, с обзиром да ова променљива представља одређени регистар унутар FPGA чипа чија

је почетна вредност постављена на нулу STARTUP блоком. У том случају локални бројач *timint* се неће поредити са *timerint* већ са унапред задатом вредношћу 00002000h која ће дефинисати периоду прекида.

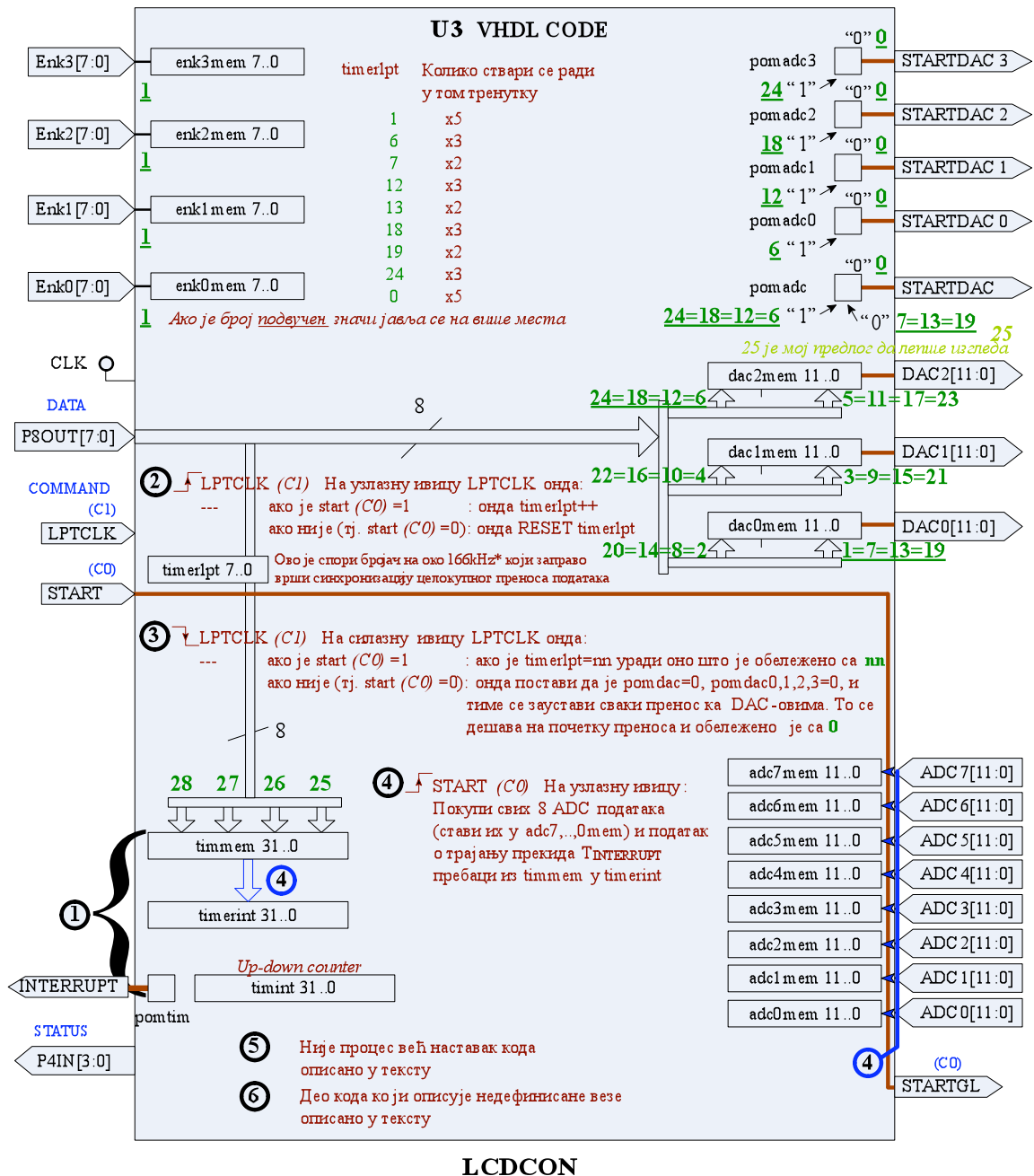
Као што се види променљива *pomtim* се директно води на линију *INTERRUPT* којом се задаје прекид. Ова променљива мења своју вредност на 1, када се вредност локалног бројача *timint* изједначи са *timerint* (односно 00002000h), и мења своју вредност на 0, када *timint* постане =0. На тај начин се добија периодичан сигнал чија периода представља периоду задавања прекида $T_{INTERRUPT}$.

Видимо да се локални бројач *timint* инкрементира кад је вредност *pomtim* =0, а да се декрементира када је *pomtim* =1. Како је периода такта FPGA чипа 125ns (Fclk = 8MHz), то ће и периода извршавања овог процеса који се јавља на његову узлазну ивицу износити 125ns. То нам казује да ће се *timint* инкрементирати, односно декрементирати на сваких 125ns. Уколико се вредност за *timerint* не задаје .C кодом, већ је периода прекида дефинисана вредношћу 00002000h то значи да ће променљива *pomint* држати вредност 1 односно 0, у периоду од 00002000h x 125ns = 8192dec x 125ns = 1.024ms, што значи да ће периода овог сигнала износити 2.048ms, односно да ће толико износити и периода задавања прекида за пренос података. У коришћеном коду вредност променљиве *perioda* којом је задата почетна вредност регистру *timerint* је постављена на 1000, што значи да је периода задавања прекида 2 x 1000 x 125ns = 250μs. Уколико би желели да повећамо брзину преноса, односно да још додатно смањимо периоду задавања прекида, могли би то остварити на тај начин што податак о трајању прекида не би преносили, већ би његову подразумевану вредност уместо на 00002000h подесили на прорачунату жељену вредност.

Слика временских дијаграма (Слика 7.16) приказује једну периода преноса у трајању од 250μs, и указује на то да је ова периода довољна за пренос свих жељених података.

2. Процес 2 се активира на узлазну ивицу C1 командног бита LPT порта *lptclk* (C1), и ако је у том тренутку вредност командног бита C0 LPT порта *start* (C0) =1, локални бројач *timerlpt* ће се инкрементирати, док уколико је *start* (C0) ≠1 тј. =0 бројач *timerlpt* ће се ресетовати.

Сада ћемо сличном сликом попут претходне описати преостале процесе VHDL кода LCDCON блока.



Слика 7.5 Анализа 3. 4. 5. и 6. процеса из VHDL кода за LCDCON

3. Процес 3 се активира на силазну ивицу C1 командног бита LPT порта односно на силазну ивицу сигнала *lptclk*. Овај процес има за основни циљ да прихваћену вредност за DA конверзију преда одговарајућем DA конвертору. Временску синхронизацију преноса обезбеђује бројач *timerlpt* који броји од 1 до 28. Док је *start* (C0) = 1, дешава се следећи сценарио догађаја: прво се читавају 4 8-битна броја који представљају одговарајући померај (тј. позицију) са енкодера; потом се читавају са *DATA* линија LPT порта подаци за I, II и III MAX525 чип (сваки MAX525 чип садржи 4 DA конвертора), што траје 6 периода *lptclk* (2 пута пренос по 8 бита за 12-битну конверзију, и то за 3 таква DA конвертора). Ови подаци се

привремено смештају у регистре *dac1,2,3mem*, одакле се на активан ниво *startdac* (*pomdac* =1) сигнала ти подаци прослеђују ка првом DA конвертору првог, другог и трећег MAX525 чипа. У следећем такту *lptclk*, пренос се прекида уписом нуле у *pomdac* односно *startdac*, и тада се истовремено отпочиње са преносом нових 6 бајтова (3 12-битна податка) за конверзију на другим DA конверторима сваког од три MAX525 чипа, након чега се поново на идентичан начин, активирањем *startdac* сигнала изврши пренос ка DA конверторима новоприспелих података, након чега се исти процес понови још 2 пута за треће и за четврте DA конверторе унутар сваког од MAX525 чипова. Након свега се изврши и пренос 32-битног податка о трајању периоде прекида, који се смешта у регистар *timmem*, одакле се на узлазну ивицу C0 сигнала LPT порта, преписује у претходно описани регистар *timerint*.

Након свега овога изврши се пренос трајања INTERRUPT-а ($T_{INTERRUPT}$).

4. Процес 4 се активира на узлазну ивицу START (C0) сигнала LPT порта када се подаци са свих AD конвертора MAX1202 чипа препишу у *adc0,...,7mem* регистре и када се податак о трајању прекида $T_{INTERRUPT}$ препише из *timmem* у *timerint*.

5. Овај део кода, који не представља процес у VHDL-у јер није кôд који реагује на промену неког сигнала, служи за синхронизацију и пренос података са периферије ка рачунару. Подаци се шаљу као ниблови путем 4 одговарајуће статусне линије LPT порта (P4IN (*Status*)). Синхронизација преноса је извршена према вредности *timerlpt* бројача (нпр. када је *timerlpt* =1 шаље се *adc0mem* 3..0, затим за *timerlpt* =2 шаље се *adc0mem* 7..4 и тако редом).

Треба приметити да како се *timerlpt* инкрементира на узлазну ивицу *lptclk* (C1 бит LPT порта) (процес 2), онда ће се и сам пренос података (ниблова) ка рачунару вршити на узлазну ивицу *lptclk*, иако се пренос заправо не врши на ивицу већ на ниво сигнала *lptclk*.

Приликом овог преноса, врши се пренос следећих података:

<i>adc0mem</i> (12b), <i>adc1mem</i> (12b), <i>adc2mem</i> (12b), <i>adc3mem</i> (12b)	(12) – <i>timerlpt</i> 1..12
<i>enk0mem</i> (8b), <i>enk1mem</i> (8b)	(4 нибла) – <i>timerlpt</i> 13..16
<i>adc4mem</i> (12b), <i>adc5mem</i> (12b), <i>adc6mem</i> (12b), <i>adc7mem</i> (12b)	(12) – <i>timerlpt</i> 17..28
<i>enk2mem</i> (8b), <i>enk3mem</i> (8b)	(4 нибла) – <i>timerlpt</i> 29..32

Дакле *timerlpt* броји од 0 до 32, а у случају да је *timerlpt* <1 или >32 по повратним (*Status*) линијама шаље се дефинисана вредност – нула.

6. Овај део кода експлицитно дефинише преостале конекције (на Слика 7.5 - браон обојене):

```

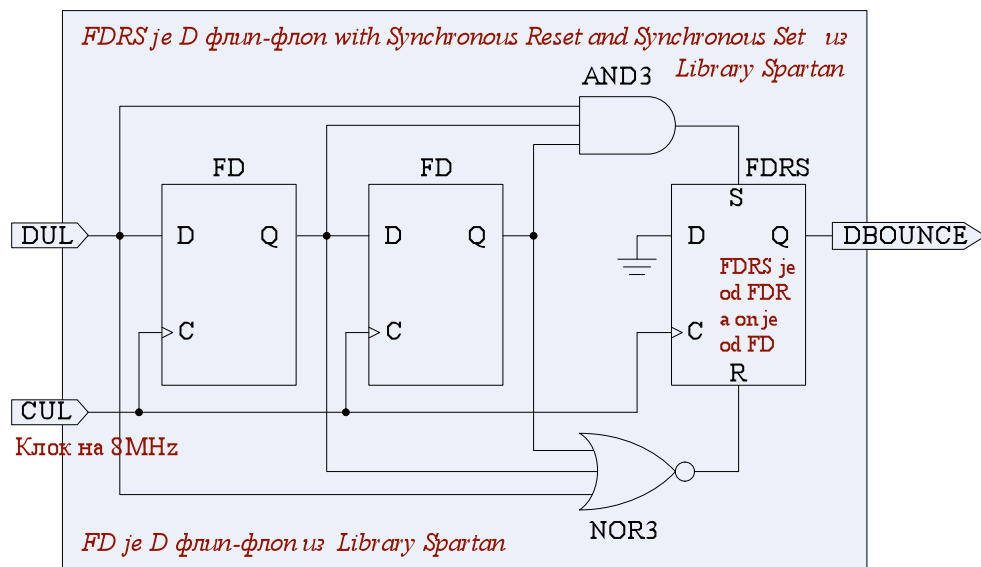
dac0,1,2 <= dac0,1,2mem;      startdac0,1,2,3 <= pomdac0,1,2,3;      startdac <= pomdac;
interrupt <= pomtim;          startgl <= start

```

Везе које су већ дефинисане претходним процесима су обележене или стрелицама или црним задебљаним линијама.

7.2.5. DEBOUNCE

DEBOUNCE је блок који је описан електричном шемом, уз коришћење понуђених стандардних логичких елемената. Шема је дата на следећој слици (Слика 7.6).

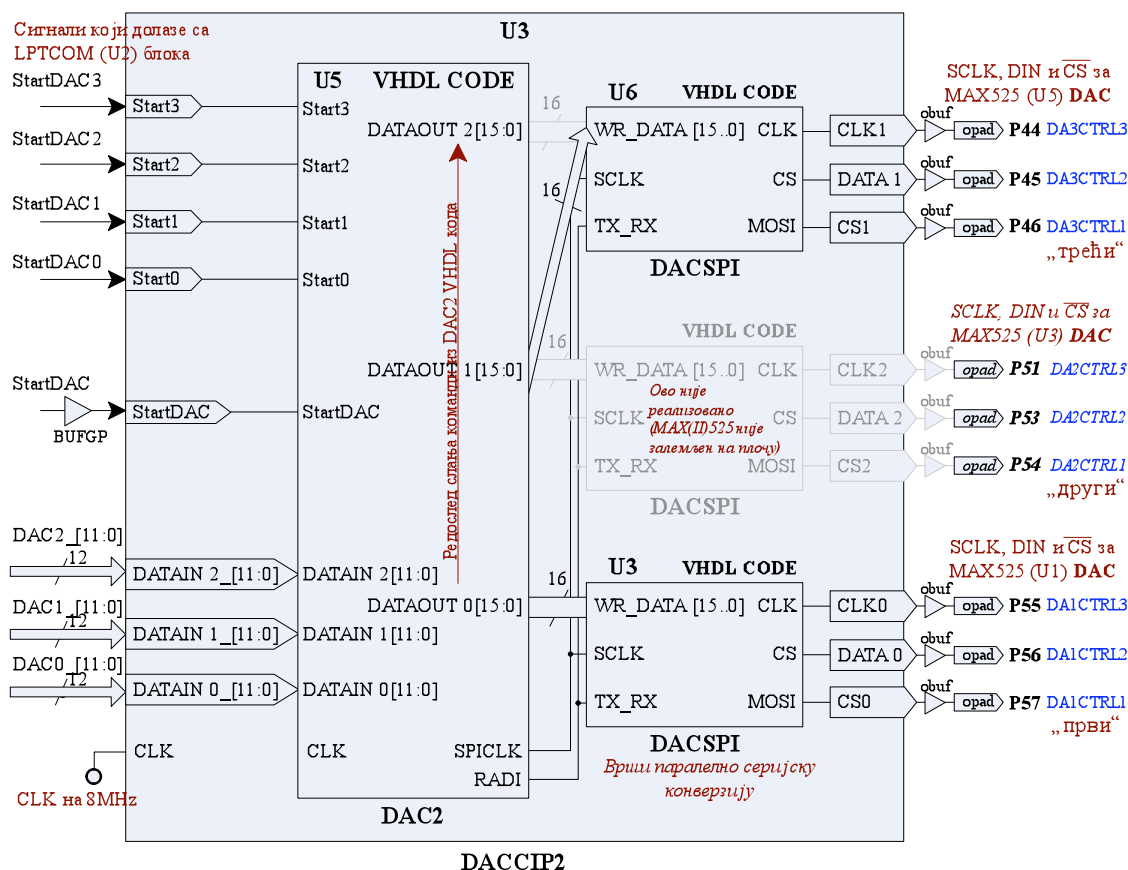


Слика 7.6 Логичка шема кола за дебаунсирање

Приказано коло за дебаунсирање је неинвертујуће и користи се за дебаунсирање командних сигнала C0 и C1. Оно заправо врши њихову синхронизацију на такт FPGA чипа од 8MHz, али и отклања гличеве краће од 2 периоде такта (краће од $2 \times 1/8\text{MHz} = 2 \times 125\text{ns} = 250\text{ns}$).

7.2.6. DACCIP2

DACCIP2 је блок који се састоји из више међусобно повезаних под блокова који су описани VHDL кодом. Ти под блокови су DAC2 и више блокова DACSPI. Њихова логичка међувеза дата је на следећој слици (Слика 7.7). Глобална функција DACCIP2 блока је да податке и команде за DA конверзију прилагоди интерфејсу и проследи одговарајућим MAX525 чиповима.



Слика 7.7 Логичка шема DAC2 блока

Блок DAC2 припрема команде и податке за DA конверзију i -тог DA конвертора унутар свакога од три MAX525 чипа. Како на самој плочи „други“ MAX525 чип није залемљен, то је овде начињено „укрштање“ (које није неопходно!) тако да се подаци намењени за „други“ MAX525 чип, прослеђују „трећем“, док се подаци намењени за „трећи“ чип, који су спремни у DAC2 блоку не прослеђују даље.

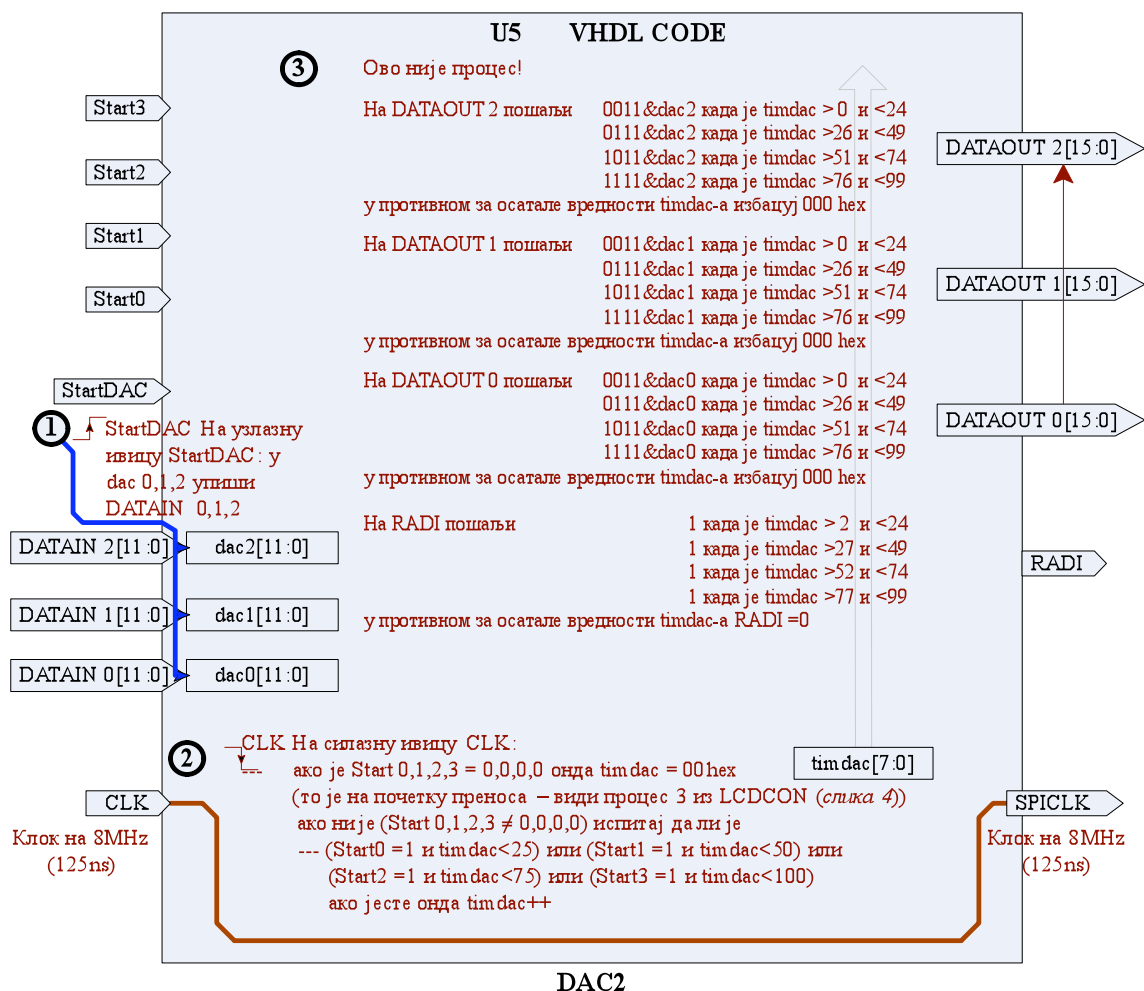
На слици је назначено, сивим линијама, шта би требало додати у DAC2 шематику да би радио и трећи, тј. „други“ MAX525 чип, уколико би се залемлио на плочу. Назначене линије указују на то, да би у том случају било отклоњено укрштање линија, односно први послати податак, за i -ти DA конвертор, би се прослеђивао „првом“, други „другом“, а трећи „трећем“ MAX525 чипу.

DAC2

7.2.7. DAC2

DAC2 је блок чија је функција описана VHDL кодом. Он прима податке са LCDCOM из LPTCOM блока, и пакује их у 16 битне податке у којима се налазе спојени команда и податак за конверзију које потом DACSPI блок серијски прослеђује одговарајућем MAX525 чипу.

Функција овог блока детаљно је описана следећом сликом (Слика 7.8).



Слика 7.8 Анализа 1. и 2. процеса и 3. дела кода из VHDL кода за DAC2

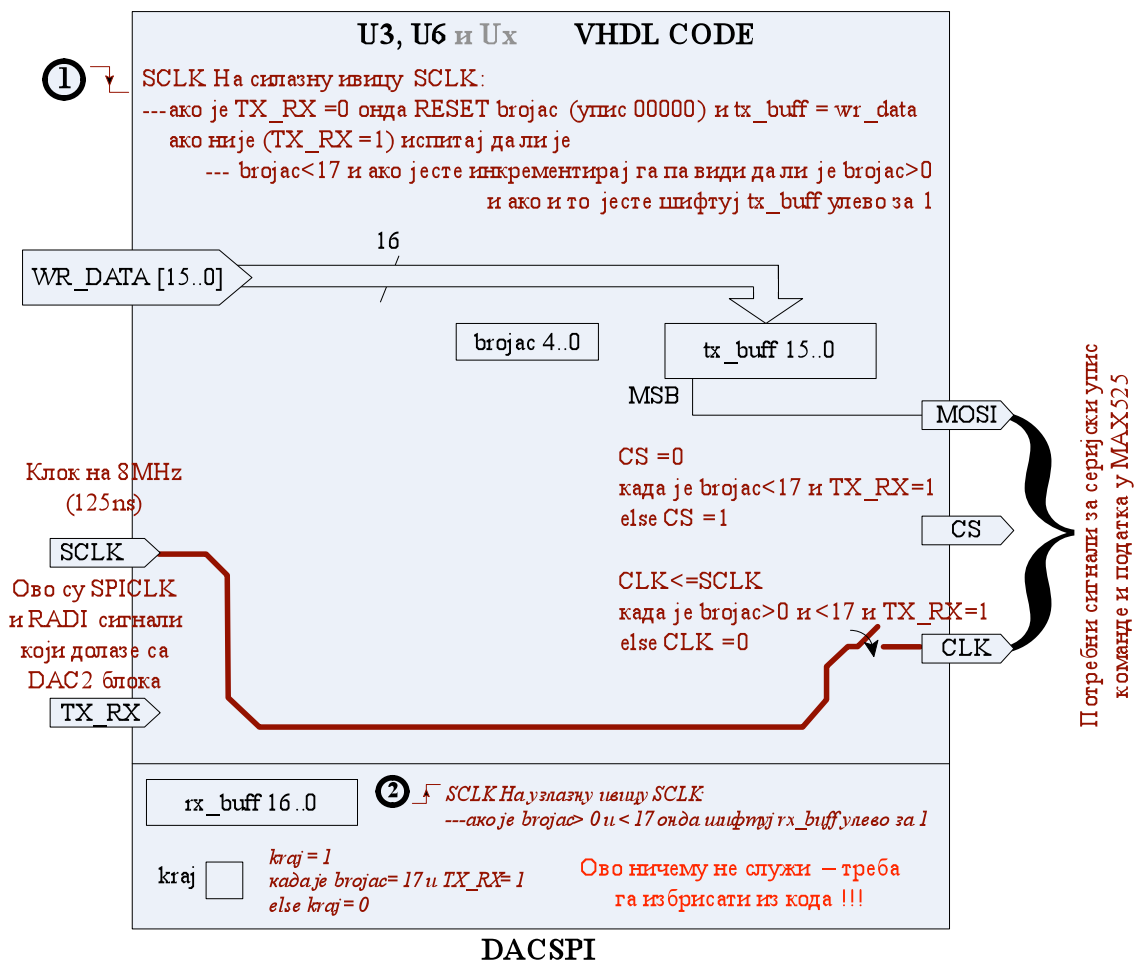
1. Процес 1 преписује спремљене вредности за DA конверзију из LCDCON блока (LCDCON блок је блок унутар LPTCOM блока) у 12-битне регистре dac0, dac1 и dac2, на улазну ивицу StartDAC сигнала.

2. Процес 2 дефинише понашање бројача *timdac* који обезбеђује синхронизацију преноса података из овог блока у DACSPI блок. Овај процес реагује на силазну ивицу такта FPGA чипа на начин како је то описано на слици 7, а изглед овог сигнала је приказан на графику временских дијаграма (Слика 7.16).

Видимо да када се испуне одговарајући услови *timdac* ће се инкрементирати 25 пута, што ће трајати $25 \times 1/8\text{MHz} = 25 \times 125\text{ns} = 3.125\mu\text{s}$.

3. Овај део кода формално не представља процес у VHDL програмском језику. У овом делу кода се врши припрема и пренос података из DAC2 блока у DACSPI блок. Синхронизација преноса обезбеђена је *timdac* бројачем из претходног процеса 2. Такође запажамо да спремљени подаци остају $(25-2) \times 125\text{ns} = 2.875\mu\text{s}$ на линијама DATAOUT0,1,2 [15:0].

Функција DACSPI блока је описана VHDL кодом. Овај блок паралелним преносом добија 16 битне податке који представљају команду и податак за DA конверзију, а затим их серијским путем шаље ка одговарајућем DA конвертору. Сада ће та функција бити сликовито објашњена (Слика 7.9), а изглед одговарајућих сигнали и садржај одговарајућих регистра приказан је на Слика 7.16.

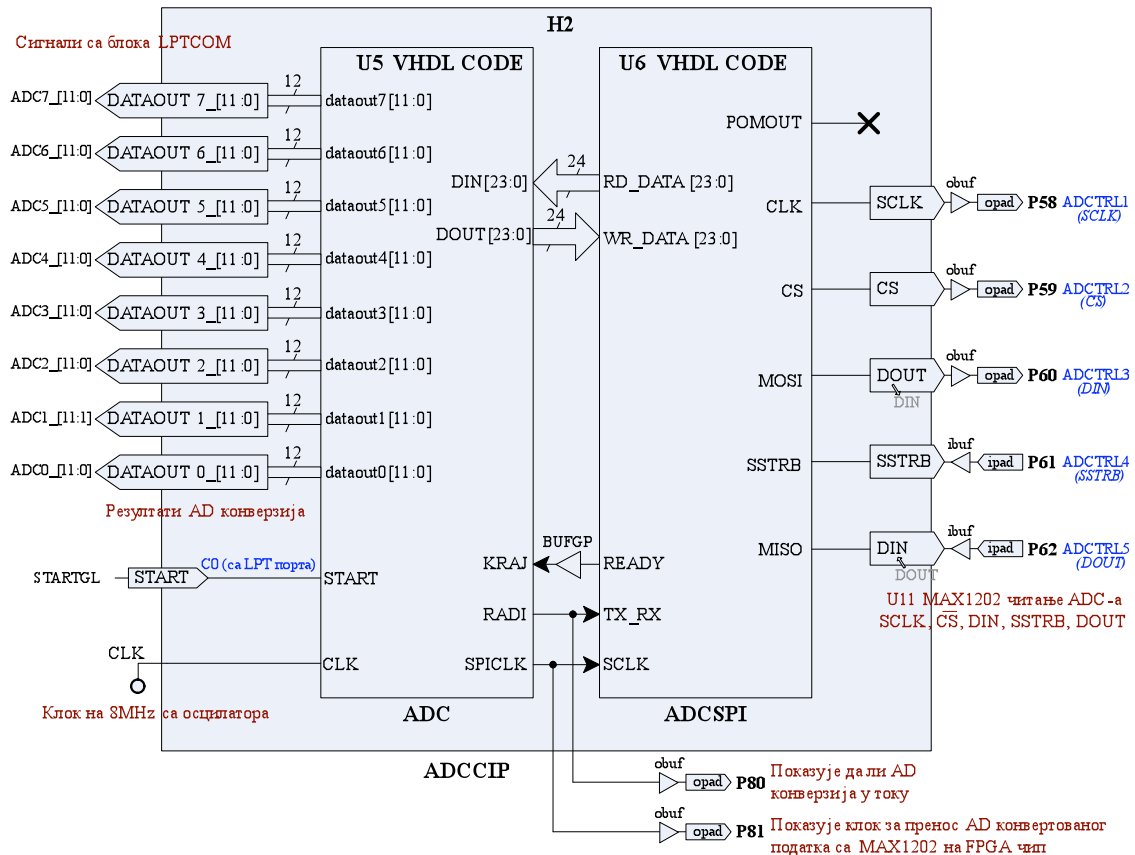


Слика 7.9 Анализа VHDL кода за DACSPI

1. Процес 1 реагује на силазну ивицу SCLK сигнала (сигнал такта на 8MHz). Овај процес врши препис података са излазних линија DATAOUT1,2,3 у 16-бини tx_buff shift регистар за серијско слање података. Када је вредност сигнала TX_RX једнака 0, односно када нема преноса сигнала, тада се ресетује вредност 5-битног бројача brojac, који има улогу да када сигнал за слање TX_RX постане активан, потпомогне синхронизацију и пребројавање послатих података. Овај бројач обезбеђује да се селекује одговарајући MAX525 чип (CS=0), а да се потом на наредних 16 сигнала такта CLK, путем линија MISO, шифтовањем tx_buff регистра, изврши пренос команде и податка из FPGA чипа ка одговарајућем MAX525 чипу.

7.2.9. ADCCIP

ADCCIP је блок који обезбеђује прилагођење централног блока FPGA чипа – LPTCOM, интерфејсу AD конвертора MAX1202. ADCCIP блок се састоји из више међусобно повезаних делова који су описани VHDL кодом. Међувеза тих блокова приказана је следећим шематиком (Слика 7.10).

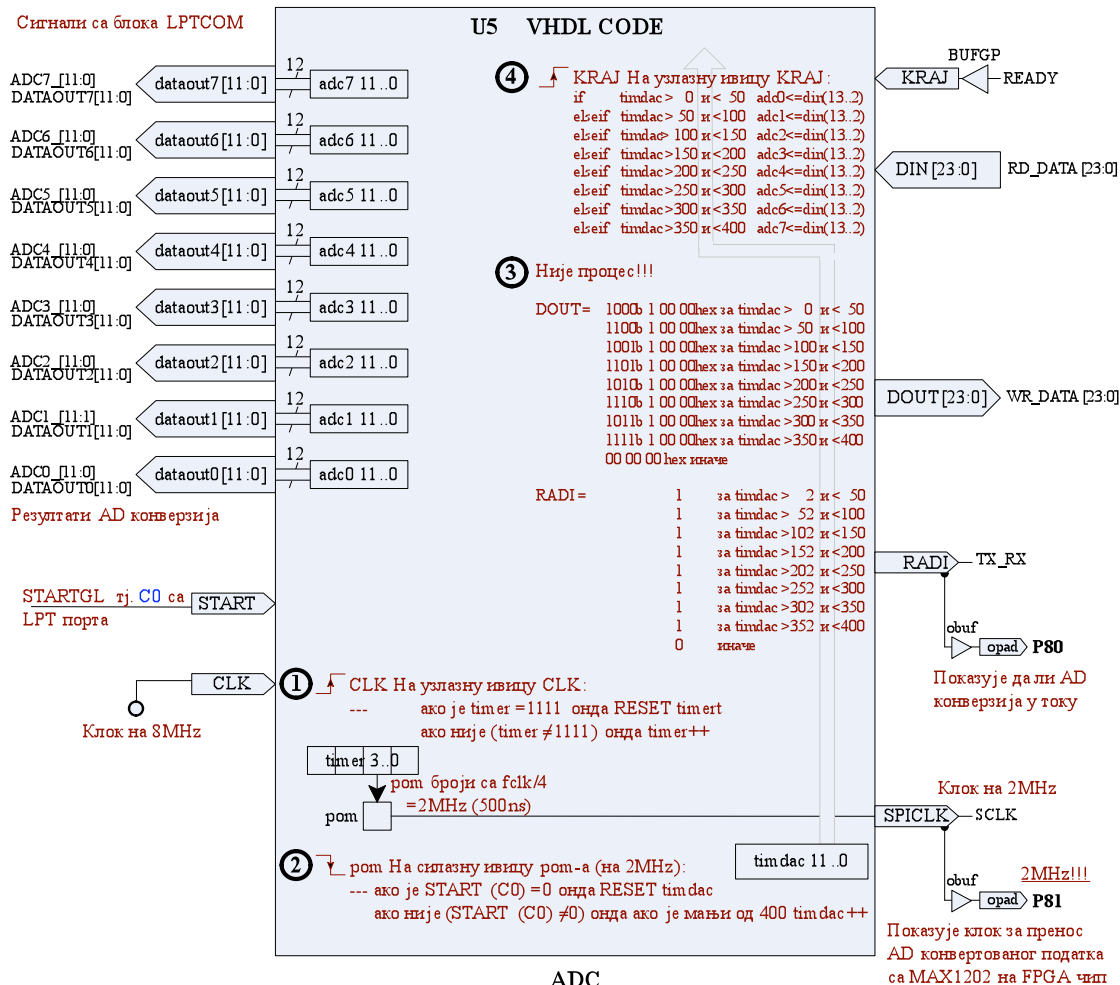


Слика 7.10 Логичка шема ADCCIP блока

7.2.10. ADC

ADC **П**

ADC је блок чија је функција описана VHDL кодом. Сада ће та функција бити сликовито и текстуално описана (Слика 7.11).



Слика 7.11 Анализа VHDL кода ADC блока

1. Процес 1 на узлазну ивицу инкрементира кружни 4-битни бројач timer. Помоћна променљива rom која представља први бит овог регистра периодично мења своју вредност са фреквенцијом од 2MHz, и служи као такт за пренос података са AD конвертора ка FPGA чипу.

2. Процес 2 обезбеђује синхронизацију процеса преноса података, помоћу 12-битног бројача timdac, који реагује на силазну ивицу помоћног такта на 2MHz (rom) и који броји од 0 до 400. Промена садржаја регистра timdac приказана је на Слика 7.16.

3. Овај део кода формално не представља процес у VHDL-у. Улога овог дела кода је да на излазну 24-битну магистралу DOUT[23:0] постави податак, који заправо представља команду и податак за AD конверзију на MAX1202 чипу (детаљније о формату команде и податку који се шаље ка AD конвертору видети у поглављу 3.3.4 о MAX1202 чипу). Временска синхронизација, када се који податак шаље ка конвертору, остварена је на основу тренутне вредности локалног бројача timdac. Овај део кода, такође на основу истог бројача, генерише и сигнал RADI (TX_RX), који казује када је читање AD конвертора у току.

ADCSPI

ADCSPI је блок који врши серијско читавање резултата AD конвертора и паралелни пренос добијених података до претходно описаног, ADC блока. Овај блок је описан VHDL кодом, а на следећој слици ће бити сликовито описана функција коју тај код имплементира (Слика 7.12).



Синхронизација преноса је остварена помоћу локалног бројача – бројас. TX_RX сигнал (сигнал RAD1 са ADC блока), казује да ли је пренос у току. Када је пренос у току, прво се врши селекција чипа (MAX1202), и чип је селектован све време док је вредност бројача од 1 до 25. Након што је селектована периферија, са кашњењем од једне периоде такта пушта на излаз и сигнал такта CLK (дакле вредност бројача је 2, 3, 4, ..., 25 – што траје $24 \times T_{SCLK}$).

1. Процес 1 се активира на силазну ивицу сигнала SCLK. Када пренос није у току (TX_RX=0), овај процес врши упис података, спремлених од стране ADC блока, у бафер за серијско слање података – tx_buff 23..0.

Када отпочне пренос (TX_RX =1), бројач почиње са бројањем, и броји од 0 до 26 (на силазну ивицу SCLK, 2MHz). Тада се врши и слање 24-битног податка (команда + нуле) ка AD конвертору по линији MOSI, шифтовањем tx_buff регистра.

2. Процес 2. На улазну ивицу, у истом периоду док се бројач мења од 2 до 25, врши се и пријем података са AD конвертора, преко линије MISO и смештање тих података у rx_buff шифтовањем улево. Када пренос није у току у rx_buff регистар се уписују нуле.

Запазити да се инкрементирање бројача и слање података, бит по бит, ка контролном регистру (1 бајт) AD конвертора врше на силазну ивицу SCLK такта, док се пријем врши на улазну ивицу истог сигнала. То је урађено тако, јер се при сваком примљеном биту у контролни бајт AD конвертора, истискује и шаље натраг истиснути бит, а њега је могуће прихватити тек након одређеног времена, тј. пола периоде, на улазну ивицу такта, путем линије MISO.

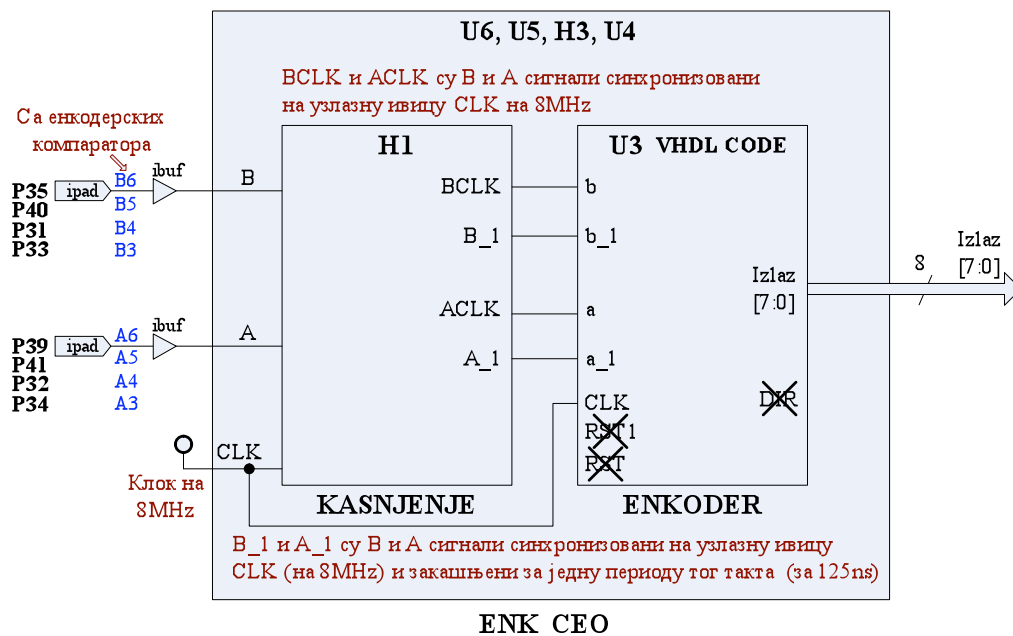
И по извршеном слању података ка AD чипу, док је још увек вредност TX_RX =1, извршиће се инкрементирање бројача на вредност 26, и чак ће се извршити и шифтовање tx_buff регистра (*else* део процеса 1), али то AD чип неће прихватити (уписати) с обзиром да неће постојати такт који би извршио упис тог податка. Потом ће се иста ситуација десити још једном када ће *brojac* добројати до вредности од 27 и извршити још једно шифтовање, које ће поново бити игнорисано од стране AD конвертора. Тада ће се генерисати и сигнал KRAJ (*brojac*=27 и TX_RX=1).

За потпуно разумевање преноса, погледати и график временских дијаграма (Слика 7.16).

7.2.12. ENK_CEO

ENK_CEO је блок који врши пријем фаза енкодерских сигнала и на основу њих израчунава позицију вратила коју прослеђује LPTCOM блоку у виду 8-битног броја.

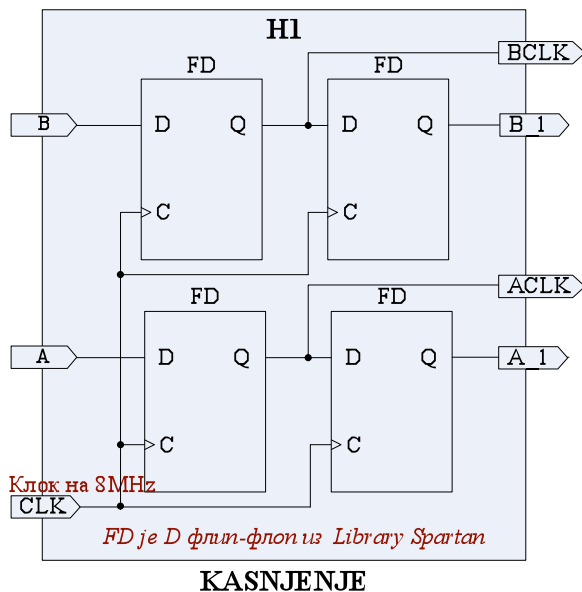
Овај блок се састоји од два под блока, првог KASNJENJE, који врши синхронизацију улазних сигнала са тактом FPGA чипа на 8MHz, и другог ENKODER, који на основу ових синхронизованих сигнала израчунава позицију. Први блок је реализован шематик фајлом, док је други реализован VHDL кодом. Логичка међувеза ових блокова приказана је на следећој слици (Слика 7.13).



Слика 7.13 Логичка шема ENK_CEO блока

7.2.13. KASNJENJE

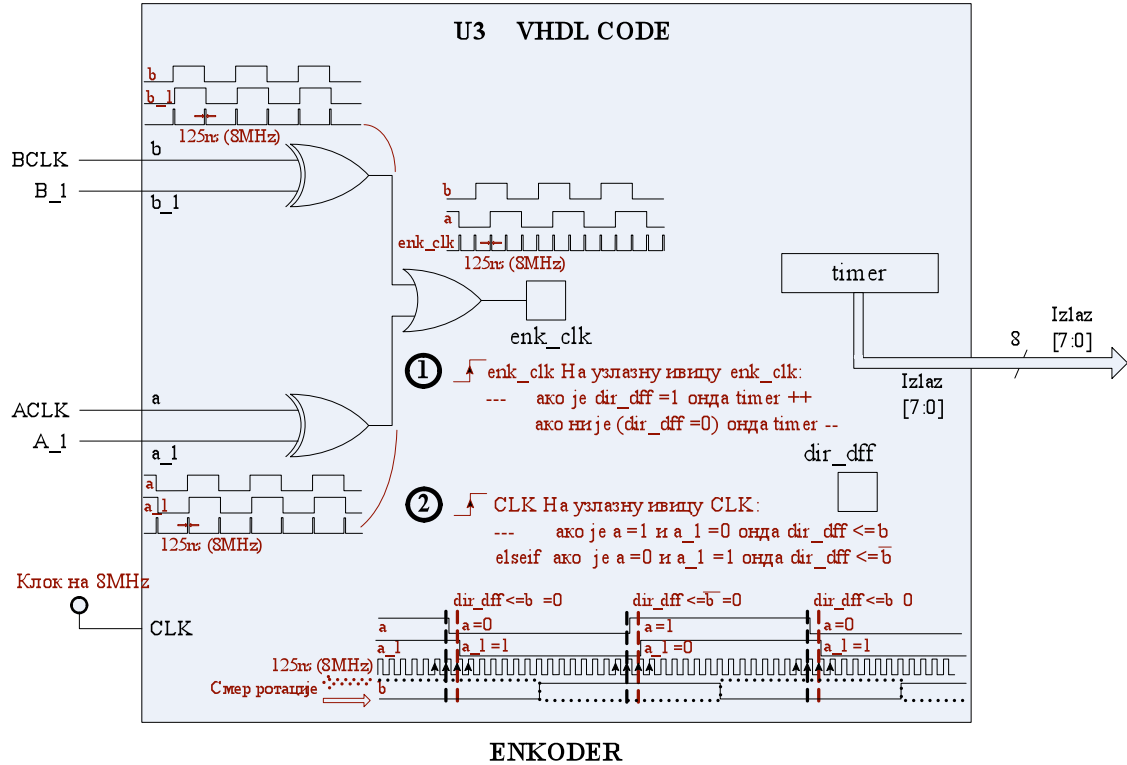
KASNJENJE је једноставан блок који врши синхронизацију улазних сигнала, који представљају фазе енкодера, са тактом FPGA чипа, који ради на 8MHz (Слика 7.14). С обзиром да се D флип-флопови тактују на улазну ивицу, то се и синхронизација ових сигнала врши на улазну ивицу CLK сигнала. Синхронизовани сигнали су обележени са BCLK и са ACLK. Такође овај блок врши и кашњење ових сигнала (BCLK и ACLK) за једну периоду такта односно за $(1/8\text{MHz} = 125\text{ns})$, и ти сигнали су обележени са B_1 и A_1.



Слика 7.14 Логичка шема блока KASNJENJE

7.2.14. ENKODER

ENKODER је блок чија је функција описана VHDL кодом. На следећој слици је та функција сликовито описана (Слика 7.15).



Слика 7.15 Анализа VHDL кода ENKODER блока

Са претходне слике се види да се сигнал *enk_clk* јавља 4 пута у периоди сигнала који долази са енкодера, тј. јавља се и на узлазну и на силазну ивицу обе фазе енкодера. Тиме је прецизност мерења повећана 4 пута.

1. Процес 1. На сваку узлазну ивицу овог сигнала (*enk_clk*) проверава се стање флип-флопа *dir_dff*, који представља смер ротације мотора и у зависности од смера, инкрементира се или декрементира бројач *timer* који представља позицију вратила мотора.

2. Процес 2 служи за одређивање смера ротације мотора (*dir_dff*). На временском дијаграму (при дну Слика 7.15) који описује процес 2, поредећи узлазне ивице фаза *a* и *b* енкодера, можемо рећи да, у приказаном примеру, фаза *a* касни за фазом *b*. То се физички дешава при једном одређеном смеру ротације мотора. У случају супротног смера ротације временски распоред сигнала *a* и *b* биће обрнут. Са слике се види да ће за приказани смер ротације вредност регистра *dir_dff* који описује смер бити нула. Лако се може приметити какав се резултат добија при промени смера ротације. Тада ће, као што је речено, узлазна ивица фазе *b* каснити за узлазном ивицом фазе *a*. То најједноставније можемо посматрати тако што ћемо на посматраном временском дијаграму уместо сигнала *b* нацртати

инвертовани сигнал (тачкаста линија). Тада се јасно види да је вредност *dir_dff* флип-флопа променила вредност, односно једнака је један, и указује на супротан смер ротације мотора.

Са Слика 7.5 видимо да се 8-битни излаз који представља позицију вратила прослеђује LCDCON (блоку унутар LPTCOM блока), чији процес 3, врши пријем ових података на прву силазну ивицу LPTCLK такта (уколико је и *start* (C0) сигнал =1) у одговарајући *enk0,1,2,3mem* регистар. Пренос података из ових регистара се потом врши у петом делу (није процес) са слике 4, који врши пренос ових података, путем статусних линија LPT порта, ка рачунару. Подаци се преносе нибл, по нибл, и овај пренос се практично врши на узлазну ивицу LPTCLK (C1) сигнала, што се детаљније може видети са временских дијаграма са Слика 7.16.

7.2.15. Временски дијаграми

На следећој слици (Слика 7.16), су приказани сви релевантни сигнали и одговарајуће вредности регистара из претходно описаних блокова. Слика садржи легенду која нас упућује на то у којем блоку се налазе сигнали и регистри из тог дела графика. Може се приметити да су приказани само сигнали и регистри из блокова који су описани VHDL кодом. То и јесте логично, с обзиром на то да блокови описани шемама или су јако једноставни па су прикладно описани текстуално (нпр. блок за дебаунсирање или блок за кашњење), или су блокови који описују међувезе осталих блокова (тако да ту нема регистара, а сигнали који, наравно постоје, су улази или излази блокова који су описани VHDL кодом па су гледано са стране тих блокова нацртани и означени на графику – Слика 7.16). Треба приметити да на Слика 7.16 недостаје један од блокова описаних VHDL кодом, и то је блок *ENKODER*, али његова улога и временски сигнали су описани при самом његовом објашњењу (поглавља 7.2.12, 7.2.13, 3 и 7.2.14), а временски пренос његовог продукта (позиције вратила мотора) који је увек доступан блоку LCDCON (блоку унутар LPTCOM блока), описан је временским дијаграмима који описују LCDCON VHDL фајл.

Такође треба приметити да су линије које диктирају пренос, заправо две контролне линије (C0 и C1) које се шаљу програмски (из .С кода) путем контролних линија LPT порта. На самом графику (Слика 7.16) постоје и одређени сигнали који су временски непропорционални осталим, а то су сигнали брзог такта на 8 и на 2 MHz. То је учињено, да би графици били прегледнији. На свим тим местима је додатно прокоментарисано то одступање, и наведено колико ту заправо има сигнала такта, или колико је пута тај сигнал бржи у односу на неки други.

Приказани график представља временске дијаграме за FPGA чип који би могао користити све ресурсе на плочи. Треба приметити да је временски ограничавајући фактор брзина такта *LPTCLK* (C1) сигнала, која зависи од самог .С кода, али и од брзине магистрале рачунара. У приказаном примеру (за рачунар на којем су вршена тестирања) периода *LPTCLK* (C1) сигнала износи око 6 μ s, па је за извршење целокупног преноса потребно око 32x6 μ s =192 μ s, плус две периоде такта (12 μ s) за отпочињање процеса преноса. То говори да је укупно потребно време за пренос података за све жељене ресурсе који се налазе на прилагодено-комуникационој картици око 204 μ s. Међутим ради сигурности и времена потребног рачунару да прихвати прекид који од њега тражи картица, узето је да је време потребно за један целокупни пренос података 250 μ s (што је задато .С кодом), па је за тај случај и нацртан график (Слика 7.16).

7.3. ПРИЛОГ С – Списак компоненти прилагодно-изолационе картице

У наставку је дат списак компоненти прилагодно-изолационе картице генерисан у програмском пакету OrCAD Capture.

Item	Quantity	Reference	Part
1	24	C1,C3,C5,C6,C13,C14,C15, C19,C21,C23,C24,C28,C29, C30,C35,C36,C37,C41,C43, C44,C45,C46,C47,C52	100n
2	1	C2	10u
3	3	C4,C7,C11	100u
4	6	C8,C17,C26,C33,C40,C50	10n
5	12	C9,C12,C18,C20,C22,C27, C31,C34,C38,C42,C48,C51	15p
6	1	J1	CONN DSUB 25-R
7	1	J2	CONN DSUB 37-P
8	1	J3	CONN DSUB 25-P
9	6	R1,R18,R35,R50,R65,R83	47
10	6	R2,R3,R12,R20,R22,R31	100K
11	6	R4,R25,R38,R58,R76,R94	39
12	4	R6,R7,R26,R27	2K7
13	6	R8,R23,R42,R55,R67,R86	1K
14	6	R9,R24,R43,R56,R68,R87	120 / 2W
15	12	R10,R16,R28,R33,R39,R44, R59,R61,R75,R79,R95,R98	33K
16	2	R11,R30	6K8
17	6	R15,R19,R49,R51,R80,R84	330
18	8	R37,R48,R57,R64,R71,R82, R90,R100	120K
19	4	R40,R53,R69,R88	390K
20	8	R41,R47,R54,R63,R70,R81, R89,R99	47K
21	4	R46,R62,R78,R97	10K
22	2	R72,R91	470
23	4	R74,R77,R93,R96	220
24	1	U1	7805
25	6	U2,U6,U8,U11,U12,U16	HCPL-7840
26	2	U3,U7	OP-07
27	2	U4,U15	74HC04
28	4	U5,U10,U14,U17	HCPL2631
29	2	U9,U13	TS272
30	1	U18	ULN2003A

7.4. ПРИЛОГ D – Упознавање са софтвером

Програм који врши покретање асинхроног мотора написан је у програмском језику C. Тренутно код омогућава само задавање синусоида одређених амплитуда и учестаности које командују контролним струјама i_d и i_q мотора, и прихватање енкодерског сигнала односно позиције ротора мотора.

Програм се састоји од 3 модула `GLAVNI.C`, `PORT.C` и `REGULTR.C`. Основни модул је `GLAVNI.C` и у њему се поред функције `int main()`, налази још пар дефиниција одређених функција које се пре свега тичу рада са ресурсима РС рачунара. Функције које су дефинисане у овом модулу су:

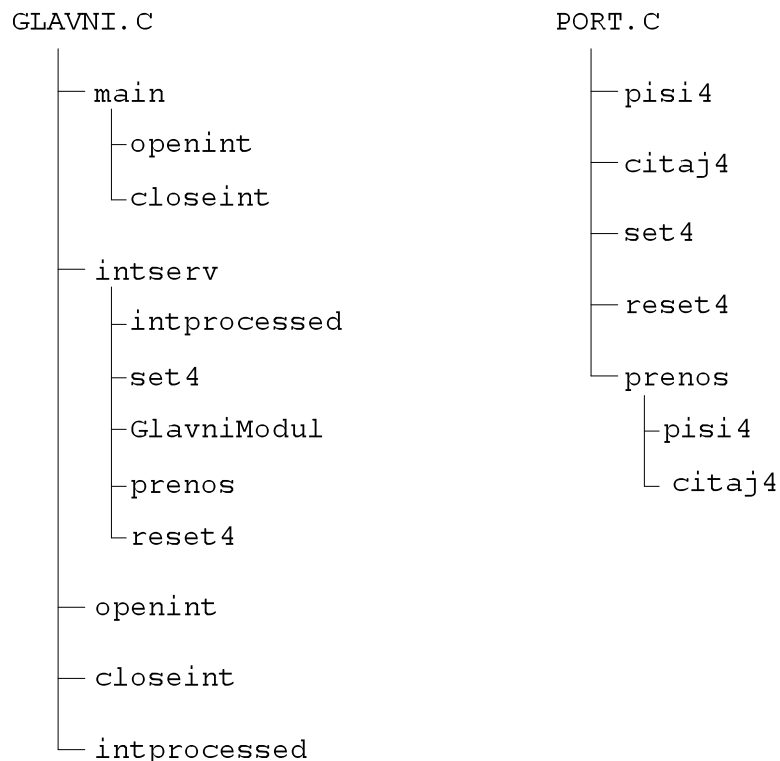
1. `void interrupt far intserv(void)` – функција саме прекидне рутине
2. `void openint(void)` – функција врши доделу нове прекидне рутине прекидном захтеву паралелног порта
3. `void closeint(void)` – ради супротно од `openint` и враћа „стару“ прекидну рутину штампача
4. `void intprocessed(void)` – функција која сигнализира да је прекид прихваћен.

Други модул је `PORT.C` и у њему су дефинисане функције које се тичу преноса података, и то су:

1. `void pisi4(unsigned char ul)` – функција врши упис контролне речи у контролни регистар
2. `unsigned char citaj4(void)` – чита нибл података из статусног регистра
3. `void set4(void)` – сетује трећи бит у контролном регистру
4. `void reset4(void)` – ресетује трећи бит у контролном регистру
5. `void prenos(void)` – врши пренос података.

Трећи модул је `REGULTR.C` и у њему се врши обрада приспелих података и припрема нових за слање. Ово је заправо део кода који у будућности треба мењати и проширивати.

На следећој слици (Слика 7.17) приказана је структура кода.



Слика 7.17 Структура коришћеног кода

У наставку ће бити дат кратак опис свих коришћених функција, али ће се настојати да буду објашњене у оном смеру тако да се прво објасне функције које не позивају друге, а потом оне које позивају друге функције, тј. оне које су претходно већ објашњене. Сходно том правилу објашњење функције саме прекидне рутине `void interrupt far intserv(void)`, која позива доста других функција ће бити одложено за пред крај, док ће објашњење функције `int main()` уследити тек на самоме крају.

- **opensint**

Функција `void opensint(void)` прво врши забрану прекида (2) и на тај начин се осигурава да ће наредна важна секвенца команди (3-6) бити извршена баш тим редом без прекида. Затим се секвенцијално извршавају следеће команде: показивачу `oldfunc` се додељује вредност почетне адресе прекидне рутине за прекид `intlev` (3), чија је вредност `0fh`, а који у IVT табели прекида РС рачунара одговара прекиду `IRQ7` који генерише паралелни порт рачунара (LPT1). Затим се истом типу прекида `0fh` додели почетна адреса нове прекидне рутине **`intserv`** (4), и са наредне две команде (5 и 6) ресетује највиши 7. бит у регистру маске IMR (*Interrupt Mask Register*), који се

```

void opensint( void)
{
(1)  int int_mask;
(2)  disable();
(3)  oldfunc = getvect( intlev);
(4)  setvect ( intlev, intserv);
(5)  int_mask = inportb( 0x21);
(6)  outportb( 0x21, int_mask & ~0x80);
(7)  enable();
}

```

налази на адреси 21h I/O адресног простора, и на тај начин је дозвољен прекид за LPT порт (наравно прекид ће се прихватити тек у случају да је и 4. бит у самом CONTROL регистру LPT порта једнак „1“). По извршењу претходне важне секције прекиди се поново дозвољавају (7).

Сам назив функције **openint** потиче од тога што она „отвара“, дозвољава прекиде са LPT порта.

Користи се на самом почетку **main** функције када желимо да одобримо прекиде за нашу периферију – прилагодено-комуникациону картицу.

- **closeint**

Функција `void closeint(void)` има супротну улогу од функције **openint**. Она врши враћање „старе“ (*default*) рутине за штампач одговарајућем прекиду типа 0fh. Поново, као и у функцији **openint** на почетку се забрањују прекиди (2), а на крају се поново дозвољавају (6) да процес рестаурације IVT табеле (3-5) не би био погрешан. Дакле, сада се типу прекида 0fh (intlev је променљива чија је вредност 0fh) враћа натраг стара вредност, сачувана у oldfunc (3), а затим се са наредне две инструкције (4 и 5) сетује 7. бит у IMR регистру маске прекида и на тај начин се поново забрањују прекиди који долазе са серијског порта (без обзира на вредност 4. бита CONTROL регистра LPT порта).

```
void closeint( void)
{
(1)  int int_mask;
(2)  disable();
(3)  setvect ( intlev, oldfunc);
(4)  int_mask = inportb( 0x21) | 0x80;
(5)  outportb( 0x21, int_mask);
(6)  enable();
}
```

Назив функције **closeint** потиче од чињенице да ова функција „затвара“, забрањује прекиде који долазе са LPT порта.

Користи се на самом крају **main** функције када желимо да забранимо прекиде периферији – прилагодено-комуникационој картици и вратимо табелу прекида (IVT) у првобитно (*default*) стање.

- **intprocessed**

Функција `void intprocessed(void)` уписује 20h („0010 0000“) у регистар који управља прекидима – ICR (*Interrupt Control Register*) и који се налази на I/O адреси 20h (1). Пети бит овог регистра је EOI (*End Of Interrupt*) и њему је додељена вредност „1“, што значи да је прекид прихваћен и да почиње његово опслуживање.

```
void intprocessed( void)
{
(1)  outportb( 0x20, 0x20);
}
```

Назив функције је кованица од *Interrupt* и *Processed*, што значи да је прекид прихваћен.

Ова функција се позива на самом почетку функције прекидне рутине **intserv**, да каже рачунару да је прекид прихваћен.

- **pis4**

Функција `void pis4(unsigned char ul)` је једноставна функција која врши упис контролне речи `ul` у командни регистар CONTROL ($I/\bar{O}$ адреса 37Ah) (1). При томе се инвертују нулти, први и трећи бит речи, и сетује се четврти бит, што је усклађено са одговарајућом реализацијом улаза саме прилагодено-комуникационе картице.

```
void pis4( unsigned char ul)
{
(1)  outportb(0x37a, (ul^0x0b) | 0x10);
}
```

Сам назив функције **pis4** треба тумачити као упис команде – пиши команду.

Користи се јако често у функцији **prenos**, за задавање команде за пренос података који се преносе било од рачунара ка картици, или од картице ка рачунару.

- **citaj4**

Функција `unsigned char citaj4(void)` врши прихватање података са периферије – прилагодено-комуникационе картице, путем статусног регистра. Податак из статусног регистра се привремено смешта у помоћну променљиву `rom` (2) да би се могла извршити потребна манипулација. Како се подаци који се шаљу са периферије налазе на позицијама 7, 5, 4 и 3 статусног регистра односно варијабле `rom`, врши се њихово преуређивање и смештање у нижи нибл помоћне променљиве `rom` (3). Функција враћа као резултат управо променљиву `rom` односно прочитани нибл података (4).

```
unsigned char citaj4 (void)
{
(1)  unsigned char rom;
(2)  rom = inportb(0x379);
(3)  rom = ((rom&0x80)>>4) + ((rom&0x38)>>3) ^ 0x08;
(4)  return (rom);
}
```

Дакле функција носи име **citaj4** које јасно сигнализира да она врши читавање података, дужине 4 бита, са периферије.

Функција **citaj4**, као и претходно описана **pis4**, користи се при преносу података тј. у функцији **prenos**.

- **set4 и reset4**

Функција `void set4(void)` сетује трећи бит у контролном регистру док функција `void reset4(void)` ресетује исти бит. Улога ових функција није битна, с бзиром да код исправно ради и без њих.

- **prenos**

Функција `void prenos(void)`, врши пренос података. Сваки блок преноса података састоји се од преноса једног бајта од рачунара ка прилагодно-комуникационој картици, и од слања једног нибла података од периферије ка рачунару. За детаље о начину и редоследу преноса погледајте сам код – модул PORT.C, али такође погледати и временских дијаграма преноса података ка FPGA чипу (Слика 7.16).

- **intserv**

intserv је функција прекидне рутине која се извршава на сваки генерисани прекид од стране периферије. Она практично посива све функције из друга два модула, односно врши прерачунавање потребних вредности, а потом врши пренос података.

- **main**

Функција **main** представља једну бесконачу петљу у којој су дозвољени прекиди од стране периферије прикључене на LPT порт рачунара. У оквиру ове функције овезбеђен је и упис жељених вредности за променљиве које представљају брзину ротације мотора, и задате струјне референце i_d^* и i_q^* . Такође обезбеђено је и правилно заустављање мотора при напуштању програма.

У наставку текста дат је код који се тренутно користи за покретање мотора.

GLAVNIC

```
#include <stdio.h>
#include <bios.h>
#include <dos.h>
#include <conio.h>
#include <math.h>

#define DATA 0x0378
#define STATUS DATA+1
#define CONTROL DATA+2
#define PI 3.141592653589793

void openint( void);
void closeint( void);
void intprocessed( void);
void interrupt far intserv( void);
```

```

extern void prenos( void);
extern void GlavniModul( void);

int intlev = 0x0f;
void interrupt far (*oldfunc) ();

extern int dac0, dac1, dac2, dac3;
extern int enk0;
extern int difenk0;
extern float pozic0, tetaR, tetaS,tetaK;

extern int flag;
int ch, izlaz=0;
float wZeljeno=0;    // Referentna brzina - zadaje se na wZeljeno u kodu
float d=675, q, pom; // Ove vrednosti za d=675 i q= 783.42 su nominalne!
extern float wMerenoRPM, wProcenjeno, tetaR;
extern float KP, KI;

int main()
{
    openint();
    outportb( CONTROL, inportb( CONTROL) | 0x10);

    while (1)
    {
        if (flag==1){printf("Motor se isuvisе brzo kretao! (Wr= %5.1f (rpm)) Pritisnite 'e' za izlaz!\n",
wMerenoRPM);}
        else
        {
            printf( "id*= %4.0f  iq= %5.0f  ",d, q);
            printf( "KP= %3.1f  KI= %3.1f  ",KP, KI);
            printf("W*= %5.1f  Wr= %5.1f (rpm)\n",wZeljeno,wMerenoRPM);
        }

        if ( kbhit() )
        {
            ch = getchar();

            switch( ch )
            {
                case 'w':printf("Unesite zeljenu brzinu (rpm) \n");
                    scanf("%f",&wZeljeno);
                    if(pom > 800) {wZeljeno = 800;printf(" Limit 800 [rpm]!\n");}
                    else if(pom < -800){wZeljeno = -800;printf(" Limit -800 [rpm]!\n");}
                    else wZeljeno = pom;
                    break;

                case 'f':printf("Unesite zeljeni fluks (tj. id*) \n");
                    scanf("%f",&d);
                    break;

                case 'i':printf("Unesite novo KI \n");
                    scanf("%f",&KI);
                    break;

                case 'p':printf("Unesite novo KP \n");
                    scanf("%f",&KP);
            }
        }
    }
}

```

```

                                break;
                                case 'e': d=0; q=0; izlaz=1; // u REGULTR se javlja deljenje sa 0!
                                getchar(); // Ovi getchar-ovi moraju da bi stigao prekid i upisale se ove
nule za d i q
                                getchar();
                                break;
                                default:
                                continue;
                                }
                                }

                                if (izlaz==1) break;
                                }

closeint();
return(0);
}

void interrupt far intserv( void)
{
    disable();
    intprocessed();
    set4();
    GlavniModul();
    prenos();
    reset4();
    enable();
}

void openint( void)
{
    int int_mask;
    disable();
    oldfunc = getvect( intlev);
    setvect ( intlev, intserv);
    int_mask = inportb( 0x21);
    outportb( 0x21, int_mask & ~0x80);
    enable();
}

void closeint( void)
{
    int int_mask;
    disable();
    setvect ( intlev, oldfunc);
    int_mask = inportb( 0x21) | 0x80;
    outportb( 0x21, int_mask);
    enable();
}

void intprocessed( void)
{
    outportb( 0x20, 0x20);
}

```

PORT.C

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>

int adc0, adc1, adc2, adc3, adc4, adc5, adc6, adc7;
int dac0 = 0, dac1 = 0, dac2 = 0, dac3 = 0, dac4 = 0, dac5 = 0, dac6 = 0, dac7 = 0, dac8 = 0, dac9 = 0, dac10 = 0, dac11
= 0;
int enk0, enk1, enk2, enk3;
int difenk0;
int enk0_1 = 0;

extern long perioda;
unsigned char b1 = 0;

void prenos( void);
void set4( void);
void reset4( void);

void pisi4( unsigned char ul)
{
    outportb( 0x37a, (ul ^ 0x0b) | 0x10);
}

unsigned char citaj4 ( void)
{
    unsigned char pom;

    pom = inportb( 0x379);
    pom = ( ((pom & 0x80) >> 4) + ((pom & 0x38) >> 3) ) ^ 0x08;
    return (pom);
}

void set4( void)
{
    b1 |= 0x08;
    pisi4( b1);
}

void reset4( void)
{
    b1 &= ~0x08;
    pisi4( b1);
}

void prenos( void)
{
    int i;

    adc0 = 0;
```

```

adc1 = 0;
adc2 = 0;
adc3 = 0;
adc4 = 0;
adc5 = 0;
adc6 = 0;
adc7 = 0;

enk0 = 0;
enk1 = 0;
enk2 = 0;
enk3 = 0;

b1 &= ~0x03;
pisi4( b1);

b1 |= 0x02;
pisi4( b1);

b1 &= ~0x03;
pisi4( b1);

b1 |= 0x01;
pisi4( b1);

outportb( 0x378, (unsigned char) ( dac0));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc0 += i;

outportb( 0x378, (unsigned char) ( dac0 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc0 += (i << 4);

outportb( 0x378, (unsigned char) dac1);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc0 += (i << 8);

outportb( 0x378, (unsigned char) ( dac1 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc1 += i;

```

```

outportb( 0x378, (unsigned char) dac2);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc1 += (i << 4);

outportb( 0x378, (unsigned char) ( dac2 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc1 += (i << 8);

outportb( 0x378, (unsigned char) dac3);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc2 += i;

outportb( 0x378, (unsigned char) ( dac3 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc2 += (i << 4);

outportb( 0x378, (unsigned char) dac4);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc2 += (i << 8);

outportb( 0x378, (unsigned char) ( dac4 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc3 += i;

outportb( 0x378, (unsigned char) dac5);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc3 += (i << 4);

```

```

outportb( 0x378, (unsigned char) ( dac5 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc3 += (i << 8);

```

```

outportb( 0x378, (unsigned char) dac6);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
enk0 += i;

```

```

outportb( 0x378, (unsigned char) ( dac6 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
enk0 += (i << 4);

```

```

outportb( 0x378, (unsigned char) dac7);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
enk1 += i;

```

```

outportb( 0x378, (unsigned char) ( dac7 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
enk1 += (i << 4);

```

```

outportb( 0x378, (unsigned char) dac8);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc4 += (i);

```

```

outportb( 0x378, (unsigned char) ( dac8 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc4 += (i << 4);

```

```

outportb( 0x378, (unsigned char) dac9);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc4 += (i << 8);

outportb( 0x378, (unsigned char) ( dac9 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc5 += (i);

outportb( 0x378, (unsigned char) dac10);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc5 += (i << 4);

outportb( 0x378, (unsigned char) ( dac10 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc5 += (i << 8);

outportb( 0x378, (unsigned char) dac11);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc6 += i;

outportb( 0x378, (unsigned char) ( dac11 >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc6 += (i << 4);

outportb( 0x378, (unsigned char) perioda);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc6 += (i << 8);

```

```

outportb( 0x378, (unsigned char) (perioda >> 8));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc7 += (i);

outportb( 0x378, (unsigned char) (perioda >> 16));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc7 += (i << 4);

outportb( 0x378, (unsigned char) (perioda >> 24));
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
adc7 += (i << 8);

outportb( 0x378, (unsigned char) 0);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
enk2 += i;

outportb( 0x378, (unsigned char) 0);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
enk2 += (i << 4);

outportb( 0x378, (unsigned char) 0);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
enk3 += i;

outportb( 0x378, (unsigned char) 0);
b1 |= 0x02;
pisi4( b1);
b1 &= ~0x02;
pisi4( b1);
i = citaj4();
enk3 += (i << 4);

```

```

    b1 |= 0x02;
    pisi4( b1);
    b1 &= ~0x02;
    pisi4( b1);

    difenk0 = enk0 - enk0_1;
    enk0_1 = enk0;

    if ( difenk0 > 128)
        difenk0 -= 256;
    else if ( difenk0 < -128)
        difenk0 += 256;

}

```

REGULTR.C

```

#include <math.h>
#define PI 3.141592653589793

void GlavniModul( void);

extern int adc0, adc1, adc2, adc3;
extern int dac0, dac1, dac3, dac4;
extern int difenk0;
float pozic0 = 0;

long perioda = 1000; // Ovo je promenljiva kojom se definise Tinterrupt-a

extern float wZeljeno, d, q;
float ialfa, ibeta, Tinterrupt;
float tetaR, tetaR_staro=0, DeltaTetaR, tetaR_elektricno, tetaS=0, tetaS_staro=0, DeltaTetaS, tetaK, tetaK_staro=0,
DeltaTetaK; // DeltaTetaS i ispod wProcenjeno su samo radi ispisa
float wMereno, wMerenoRPM, wMereno_staro=0, wProcenjeno;
float Moment=0, Moment_staro=0, MOMA;
float KK=19.85, qMAX=1566; // iqMAX=2*iqNOM=2*783

float KP = 31.5, KI = 3;
int ii=1, NN=60, br_polova = 2, flag = 0; // NN služi za povećanje preciznosti merenja brzine wMereno, ii je broj koji
pomaze u tome

void GlavniModul( void)
{
    if (d!=0)
    {
        Tinterrupt = perioda/1000*290e-6; // 290e-6 je izmereno vreme prekida za perioda = 1000

        pozic0 += difenk0;
        if (pozic0>3999) pozic0-=4000;
        if (pozic0<0)pozic0+=4000;

        tetaR = pozic0*2*PI/4000;
    }
}

```

```

if (ii==NN)
{
    DeltaTetaR = tetaR - tetaR_staro;
    if (DeltaTetaR > PI) DeltaTetaR -= 2*PI;
    if (DeltaTetaR < -PI) DeltaTetaR += 2*PI;
    tetaR_staro = tetaR;
    wMereno = DeltaTetaR/(Tinterrupt*NN);           // Brzina u rad/sec
    wMerenoRPM = wMereno*60/(2*PI);

    DeltaTetaS = tetaS - tetaS_staro;           // Ova 4 naredna reda se mogu kasnije izbaciti
    if (DeltaTetaS > PI) DeltaTetaS -= 2*PI;     // Ovo služi samo za prikaz procenjene brzine
    if (DeltaTetaS < -PI) DeltaTetaS += 2*PI;
    tetaS_staro = tetaS;
    wProcenjeno = DeltaTetaS/(Tinterrupt*NN)/br_polova; // Brzina u rad/sec

    Moment = Moment_staro - KP*(wMereno - wMereno_staro) + KI*(wZeljeno*2*PI/60 - wMereno); // za
NN=34 odgovara Tinterrupt-a = 10ms
    wMereno_staro = wMereno;
    if (Moment > +qMAX) Moment = +qMAX;
    if (Moment < -qMAX) Moment = -qMAX;
    Moment_staro = Moment;
    q = Moment;
    //q = +783.42; // kada se ostavi ovaj red, posmatrana ucestanost ialfa na osciloskopu treba da je
w_klizanja_nom tj. 3.67Hz. tako se podesava KK (50:1500 => x:(1500-1390))
    ii=0;
}

ii+=1;

DeltaTetaK = KK*Tinterrupt*q/d; //na pocetku pise if (d!=0) => nikada se ne deli sa 0
tetaK = tetaK_staro + DeltaTetaK;
if (tetaK >= 2*PI) tetaK -= 2*PI;
if (tetaK < 0 ) tetaK += 2*PI;
tetaK_staro = tetaK;

tetaR_elektricno = br_polova*tetaR;
if (tetaR_elektricno >= 2*PI) tetaR_elektricno -= 2*PI;
if (tetaR_elektricno < 0 ) tetaR_elektricno += 2*PI;

tetaS = tetaR_elektricno + tetaK;
if (tetaS >= 2*PI) tetaS -= 2*PI;
if (tetaS < 0 ) tetaS += 2*PI;

ialfa = d*cos(tetaS) - q*sin(tetaS);
ibeta = d*sin(tetaS) + q*cos(tetaS);

dac0 = (int) (2073 + ialfa);
dac1 = (int) (2046 + ibeta); // trebalo bi da se dodaje 2048, ali ovo su rezultati dobijeni merenjem, koji
koriguju nelinearnost funkcije prenosa u svakoj grani po na osob
// podesavanja uradjena za d=0 i q=0 => dac0 -> 2085, a dac1 -> 2060
// ove parametre treba s vremena na vreme korigovati, a to treba uciniti
onda ako motor zuji i kada se zaustvi, to znaci da u njemu nisu ukinute struje
dac3 = (int) (2048 + q*1.73); // dac 4 -> pin 34, dac 3 -> pin 32 37-pinskog konektora
dac4 = (int) (2048 + wMereno*16); // wMereno je do 127 pa da se prosiriopseg do 4095

```

```

        if ((wMerenoRPM>1200)|| (wMerenoRPM<-1200)) // Softverska zastita
        {
            d=0; // Da ne bi prihvatao sledece interrupte
            flag=1; // Za ispis poruke upozorenja na ekranu
        }
    }

else {   dac0=2073;
        dac1=2046;
        dac3=2048;
        dac4=2048;
    }
}

```

8. ЛИТЕРАТУРА

[1] „Лабораторијска станица за испитивање алгоритама управљања индиректно векторски управљаним асинхроним мотором – ВЕКТРА“, Електротехнички факултет Београд, Београд 2001

[2] Слободан Н. Вукосавић, „Дигитално управљање електричним погонима“, Академска мисао, Београд, 2003

[3] Вујо Дрндаревић, „Персонални рачунари у системима мерења и управљања“, Академска мисао, Електротехнички факултет, Сабраћајни факултет, Београд, 2003

[4] ”Испитивање динамичких својстава дигитално регулисаног брзинског сервомеханизма путем рачунарских симулација”, Електротехнички факултет Београд, Београд 2006

[5] <http://www.xilinx.com>

[6] www.interfacebus.com/Design_Connector_1284.html

[7] www.tkk.fi/Misc/Electronics/circuits/lptpover.html