

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Viktoras Lipinskis

VĖJO JĖGAINIŲ VALDYMO SISTEMOS REALIZAVIMAS IR
TYRIMAS, NAUDOJANT FPGA MATRICAS

MAGISTRO DARBAS

Darbo vadovas: dėst. Vacius Jusas

KAUNAS, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

**VĖJO JĖGAINIŲ VALDYMO SISTEMOS REALIZAVIMAS IR
TYRIMAS, NAUDOJANT FPGA MATRICAS**

Informatikos inžinerija

Vienlustinės sistemos

MAGISTRO DARBAS

Magistrantas

Viktoras Lipinskis, IFM – 9/5

2011 m. gegužės 24 d.

Vadovas

Dėst. Vacius Jusas

2011 m. d.

Recenzentas

2011 m. d.

KAUNAS, 2011

Viktoras Lipinskis. „Vėjo jėginių valdymo sistemos realizavimas ir tyrimas, naudojant FPGA matricas“. Informatikos inžinerijos magistro baigiamasis darbas. Darbo vadovas: dėst. Vacius Jusas. Kauno Technologijos Universitetas, Informatikos fakultetas, Programų inžinerijos katedra. Kaunas, 2011. Darbo apimtis: 81 puslapių, 8 lentelių, 43 paveikslėlių.

SANTRAUKA

Šio darbo tikslas – ištirti dviejų tipų valdymo sistemas (FPGA matricos ir MCU RISC mikrovaldiklio), kurioms sukurtas vienodas vėjo jėginių valdymo algoritmas. Palyginti tyrimo rezultatus tarpusavyje bei įvertinti sistemų teigiamas arba neigiamas savybes vėjo energetikos valdymo technologijų segmente. Išnagrinėjus tyrimo rezultatus, apibrėžti optimalią valdymo sistemą, skirtingo tipo vėjo elektrinių (namų ūkio ir komercinės paskirties) kontrolei vykdyti.

Darbo tiriamasis objektas – dviejų tipų vėjo jėginių valdymo sistemos, kur pirmoji sistema realizuota tiesiogiai FPGA matricoje, o antroji įgyvendinta naudojant įprastų mikrovaldiklių RISC architektūrą. Sudarius vienodus valdymo algoritmus ir atlikus testavimą naudojant tą patį taktinį dažnį (4MHz), šios dvi sistemos yra lyginamos tarpusavyje bei analizuojamos.

Darbo struktūra: magistro baigiamąjį darbą sudaro įvadas, dvi dalys, išvados ir priedai. Kiekviena darbo dalis suskirstyta į skyrius pagal nagrinėjamų klausimų pobūdį.

Pirmoje dalyje apžvelgiamos įvairios mokslinės publikacijos vėjo jėginių valdymo sistemų tematika, remiantis straipsniuose pateikta informacija, aptariamoms bendrosios probleminės sritys. Šioje dalyje taip pat aptarta vėjo jėginių valdymo sistemų rinka bei esama valdymo sistemų situacija. Pateikti vėjo jėginių valdymo sistemų tipai ir nagrinėjami aptartų problemų sprendimo būdai.

Antroje dalyje apibrėžti tiriamojo darbo metodai ir pristatyta vėjo jėginių valdymo sistemos realizacijos ir tyrimo dalies struktūra. Aprašyta tiriamojo darbo vykdymo eiga, pateikti realizuotų valdymo sistemų tipai bei techninės charakteristikos, išanalizuotas sistemose įgyvendintas valdymo algoritmas, pateiktos valdymo algoritmo būsenų diagramos bei sudaryta bendra valdymų sistemų struktūra. Pateikti bei aprašyti vėjo jėginių valdymo sistemų (realizuotų FPGA matricoje ir MCU mikrovaldiklio architektūroje) gauti modeliavimo ir sintezės rezultatai, atliktas sistemų tarpusavio palyginimas pagal gautus duomenys.

Viktoras Lipinskis. „Implementation and Research of Wind Turbines Control System Using FPGA“. The Informatics engineering master's degree final study. Research adviser: lecturer Vacius Jusas. Kaunas University of Technology, Informatics faculty, department of software engineering. Kaunas, 2011. Size of the study: 81 pages, 8 tables, 43 images.

SUMMARY

The purpose of this study – to research two types of control systems (FPGA's and MCU RISC microcontroller's) with created control algorithm for wind turbines. To compare results of this study and evaluate advantage or disadvantages of technologies systems for wind energy. Determine optimal solution of control system for different types of wind turbines (for household and commercial usage).

Object of the study – two types of control systems, where first one is realized directly in FPGA, the second one is implemented in microcontroller's RISC architecture. After making the control algorithm and testing, these two control systems is compared with each other using the same control algorithm and clock speed (4MHz).

Structure of the study: introduction, two main sections, conclusions and appendixes. Each section of the study is divided into chapters, depending on the nature of the problems being analyzed.

The first section overviews different memoirs about control systems of wind turbine, on the straight of these science publications – discussed problem spheres. In this part also is discussed about present situation of control systems in the nowadays market. Submitted wind turbines control system's types and issued solutions of discussed problems.

The second section determines methods of research work, presented wind turbines control systems realization and structure of study part. Described implementation throw of research work, submitted types of realized control systems and technical characteristics, analyzed control algorithm which implemented in control systems, submitted state charts of control algorithms and concluded overall structure of control systems. Submitted and described modeling and synthesis results of wind turbines control system's (realized directly in FPGA and using microcontrollers architecture), accomplished comparison of results for realized control systems.

TURINYS

ĮVADAS.....	6
1. VĖJO JĖGAINIŲ VALDYMO SISTEMŲ ANALITINĖ DALIS	9
1.1 Įvadinė dalis	9
1.2 Vėjo jėgainių valdymo sistemų reikalavimai bei sprendžiamos problemos Lietuvos ir pasaulio mastu	9
1.3 Esamų vėjo jėgainių valdymo sistemų rinkos apžvalga.....	14
1.4 Projektuojamos sistemos sprendimų būdų analizė.....	18
1.5 FPGA matricos ir MCU mikrovaldiklių architektūrų apžvalgos	21
1.5.1 FPGA matricos architektūros apžvalga.....	21
1.5.2 MCU mikrovaldiklių RISC architektūros apžvalga	32
2. VĖJO JĖGAINIŲ VALDYMO SISTEMOS REALIZACIJOS IR TYRIMO DALIS.....	38
2.1 Įvadinė dalis	38
2.2 Tiriamojo darbo vykdymo eigos aprašymas ir metodai	39
2.2.1 Tiriamojo darbo vykdymo etapai	39
2.2.2 Tiriamojo darbo metodai	43
2.2.3 Tiriamojo darbo realizacijų analizavimas	52
2.2.3.1 Vėjo jėgainės valdymo sistemos Nr.: 1 realizacija	52
2.2.3.2 Vėjo jėgainės valdymo sistemos Nr.: 2 realizacija	63
2.2.3.3 Realizuotų vėjo jėgainės valdymo sistemų rezultatų palyginimas.....	71
IŠVADOS.....	77
TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	79
LITERATŪRA.....	80
PRIEDAI.....	81

IVADAS

Darbo tema – „Vėjo jėgainių valdymo sistemos realizavimas ir tyrimas, naudojant FPGA matricas“. Temoje analizuojamos esamos vėjo jėgainių valdymo sistemos, nagrinėjamos šių sistemų mokslinės, techninės problemos bei sprendimo būdai, pristatomos dvi skirtingos valdymo sistemų architektūros (šios architektūros naudojamos darbo tyrimui atlikti), kurios realizuotos FPGA matricos pagrindu. Sukurtas valdymo algoritmas realizuotas, kuris įgyvendintas FPGA matricoje ir įprasto vėjo jėgainių valdymo sistemos MCU mikrovaldiklyje (RISC architektūroje). Darbo pabaigoje, šių dviejų sistemų tyrimo rezultatai tarpusavyje palyginami bei suformuluojamos galutinės išvados, kuriose atsispindi tiriamųjų objektų privalumai arba trūkumai bei įvertinama optimali valdymo sistema skirtingiems vėjo jėgainių tipams (t.y. namų ūkio ir komercinės paskirties vėjo elektrinėms).

Darbo temos aktualumas yra labai didelis šiuolaikinėje infrastruktūroje, nes elektros energija yra viena iš pagrindinių sudedamųjų dalių, padedanti vystyti ekonomiką plėtrai. Taigi šiuo metu sparčiai augant pasaulinei ekonomikai bei didėjant vartotojų poreikiui, reikalingi didesni elektros energijos išteklių. Paskutiniaisiais duomenimis remiantis galima teigti, kad šiuo metu pasaulyje suvartojamos elektros energijos galia siekia apie 12 TW (teravatų), o per metus galios prieaugis sudaro net 2%, taigi žmogaus gyvenimo kokybė yra tiesiog priklausoma nuo elektros energijos. Siekiant padengti šiuos augančius elektros vartotojų poreikius, dažniausiai naudojami tradiciniai elektros energijos gamybos būdai, kurie vis daugiau kenkia gamtai bei yra labai nesaugūs. Todėl tampa vis labiau aktualūs alternatyviosios energetikos šaltiniai, kurie yra neišsenkantys ir neniokojantys gamtos. Pasaulyje (tame tarpe ir Lietuvoje) vienas iš populiarių atsinaujinančių energijos šaltinių yra vėjas. Vėjo energija naudojama daugiau nei 70 pasaulio šalių, daugiausia Vokietijoje, Danijoje, Olandijoje, Anglijoje, JAV ir Ispanijoje. 2008 – 2010 m. įvairiose šalyse veikė daugiau nei 200 tūkstančių vėjo jėgainių (įrengta galia daugiau nei 150 GW), iš kurių apie 61% įrengta Europoje. Vėjo energetika labai populiari ne tik pasaulyje, bet ir Lietuvoje ši sritis yra sparčiai vystoma. Lietuvos vėjo energetikų asociacijos duomenimis 2009 metų pradžioje Lietuvoje veikė 47 vėjo elektrinės, kurių bendras galingumas siekė 54,4 MW, o šiuo metu (2010 – 2011 m.) valstybė remia 170 MW bendrosios galios vėjo elektrinių įrengimą šalyje. Be to vyriausybė išleido atsinaujinančių išteklių energetikos įstatymą, kuris apibrėžia Lietuvos atsinaujinančių energijos šaltinių rinką ir nustato, Lietuvos įsipareigojimų įvykdymą Europos Sąjungai, kad 2020 m. energija iš atsinaujinančių šaltinių sudarytų 23 proc. visos šalyje suvartojamos energijos.

Vėjo energetika užima itin svarią vietą atsinaujinančių išteklių rinkoje, nes dabartinės vėjo jėgainės kuriamos vis efektyvesnės, todėl ateityje vėjo energetika laikoma viena iš potencialiausių energijos generavimo rūšių. Taigi tobulėjant vėjo jėgainėms, keliami vis didesni reikalavimai vėjo elektrinių valdymo sistemoms. Įvairiuose publikacijose analizuojami vėjo jėgainių valdymo

algoritmai, valdymo sistemos bei jų techninės charakteristikos, sprendžiamos vėjo jėgainės darbo bei generuojamos energijos kontrolės problemos.

Pagrindinės mokslinės ir techninės problemos. Dauguma mokslinių publikacijų įvardijamos problemos siejamos su sistemų patikimumu, funkcionalumu, efektyvumu, gedimų skaičiumi ne tik vėjo elektrinės konstrukcijos dalyse, tačiau ir pačioje valdymo sistemoje. Dažniausiai esamos valdymo sistemos susideda iš kelių valdymo komponentų, kuriems priskirtos skirtingos funkcijos. Šie komponentai tarpusavyje sujungti sudaro vientisą kompleksinę ir gana sudėtingą valdymo sistemą, todėl atsiranda eilę pagrindinių problemų: didesnis sisteminių gedimų skaičius, komplikotas sistemų funkcionalumo (lankstumo) realizavimas, sudėtingų algoritmų greitaveikos didinimas, kompleksinis sistemos komponentų tarpusavio suderinamumas. Siekiant sušvelninti šių problemų kiekį ir įtaką kontroliuojamam objektui, įprastų vėjo jėgainių valdymo sistemų gamybos kaštai labai stipriai išauga.

Mokslinis naujumas ir problemų sprendimo metodas. Siekiant išvengti papildomų aparatūrinių komponentų, kurie naudojami skirtingoms vėjo jėgainių valdymo funkcijoms vykdyti, šiame darbe pristatyta valdymo sistema realizuota FPGA matricoje, kuri palyginama su standartine vėjo jėgainių valdymo sistemų architektūra. Šiuo metu FPGA matricos pagrindu realizuotos valdymo sistemos vėjo energetikos rinkoje yra naujovė, tačiau sparčiai kylanti ir randanti savo vietą, vėjo jėgainių valdymo technologijų visumoje. Tokio tipo sistemos, esamų valdiklių funkcijas (realizuotas aparatūriniu pagrindu), perkelia į programinį lygmenį. Kitais žodžiais tariant, FPGA matricos pagrindu įgyvendintos valdymo sistemų funkcijos programuojamos HDL kalba, kur programavimo rezultatai sintezuojami FPGA matricos loginių ventilių ir ryšių tarp jų lygmenyje. Taigi programiniu būdu yra sudaromos pageidaujamos valdymo algoritmo funkcijos, sintezuojant (pagal HDL išėties kodą) skirtingą loginių ventilių skaičių, tipą ir ryšius tarp jų. Tokiu būdu išlaikomas didelis sistemos lankstumas, pritaikymas, sumažinamas galimų sisteminių gedimų skaičius bei sudėtingi valdymo algoritmai vykdomi žymiai sparčiau nei įprastose valdymo sistemose.

Šio darbo tikslas – ištirti dviejų tipų valdymo sistemas (FPGA matricos ir MCU RISC mikrovaldiklio), kurioms sukurtas vienodas vėjo jėgainių valdymo algoritmas. Palyginti tyrimo rezultatus tarpusavyje bei įvertinti sistemų teigiamas arba neigiamas savybes vėjo energetikos valdymo technologijų segmente. Išnagrinėjus tyrimo rezultatus, apibrėžti optimalią valdymo sistemą, skirtingo tipo vėjo elektrinių (namų ūkio ir komercinės paskirties) kontrolei vykdyti. Šiems tikslams pasiekti keliami tokie *uždaviniai*:

- apžvelgti vėjo jėgainių valdymo sistemų nagrinėjamus reikalavimus bei sprendžiamas problemas;
- atlikti esamų vėjo jėgainių valdymo sistemų rinkos apžvalgą;
- aptarti projektuojamų – tiriamų sistemų sprendimo bei realizavimo būdus;
- apžvelgti FPGA matricos ir MCU RISC mikrovaldiklio architektūras;

- atlikti tyrimą realizuojant valdymo algoritmą tiesiogiai FPGA matricoje ir MCU mikrovaldiklio RISC architektūroje;
- pateikti darbo išvadas, apibendrinančias tyrimo rezultatus.

Darbo tiriamasis objektas – dviejų tipų vėjo jėgainių valdymo sistemos, kur pirmoji sistema realizuota tiesiogiai FPGA matricoje, o antroji įgyvendinta naudojant įprastų mikrovaldiklių RISC architektūrą. Sudarius vienodus valdymo algoritmus ir atlikus testavimą naudojant tą patį taktinį dažnį (4MHz), šios dvi sistemos yra lyginamos tarpusavyje bei analizuojamos. Atsižvelgiant į realizuotų sistemų gautus tyrimo rezultatus, sprendžiami optimalių valdymo sistemų variantai skirtingiems vėjo jėgainių tipams. Šių sistemų valdymo algoritmas sudarytas išnagrinėjus komercinių ir namų ūkio vėjo jėgainių pagrindines valdymo funkcijas. Taigi šių dviejų valdymo sistemų tyrimui atlikti naudojamos funkcijos: vėjo greičio matavimas bei valdymas pagal kritinį vėjo greitį, vėjo krypties matavimas ir vėjo jėgainės kampo tarp horizontaliosios turbinos ašies ir oro srauto krypties reguliavimas, baterijų bloko įkrovos / iškrovos kontrolė, generuojamos įtampos matavimas bei informacijos perdavimas vartotojui naudojant UART protokolą.

1. VĖJO JĖGAINIŲ VALDYMO SISTEMŲ ANALITINĖ DALIS

1.1 Įvadinė dalis

Šios analitinės dalies pirmame skyriuje apžvelgiamos įvairios mokslinės publikacijos vėjo jėgainių valdymo sistemų tematika. Remiantis straipsniuose pateikta informacija, aptariamos bendrosios probleminės sritys. Antrajame skyriuje apžvelgiama vėjo jėgainių valdymo sistemų rinka bei esama valdymo sistemų situacija. Trumpai aptariamos šalys, kuriose itin paplitusi elektros energijos gamybą naudojant vėjo elektrines. Trečiajame „Vėjo jėgainių valdymo sistemos analitinės dalies“ skyriuje pateikti vėjo jėgainių valdymo sistemų tipai, nagrinėjami aptartų problemų sprendimo būdai ir suformuluojami tiriamojo darbo keliami tikslai bei uždaviniai.

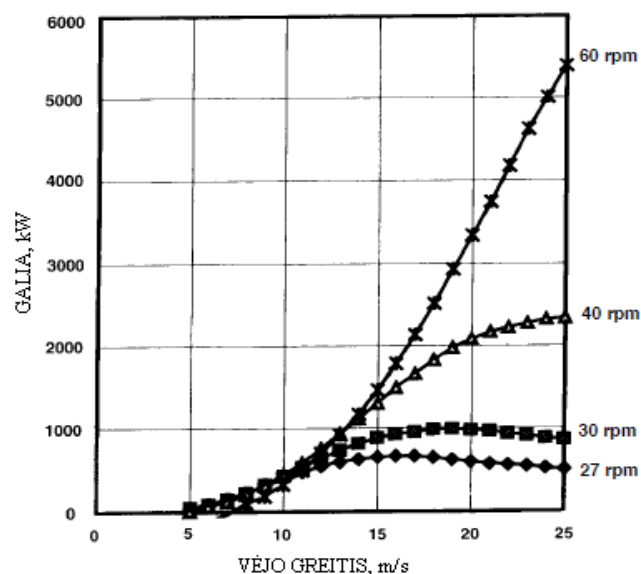
1.2 Vėjo jėgainių valdymo sistemų reikalavimai bei sprendžiamos problemos Lietuvos ir pasaulio mastu

Nagrinėjant vėjo jėgainių valdymo sistemų keliamas problemas bei reikalavimus, būtina pabrėžti, kad valdymo sistemos vienas iš pagrindinių reikalavimų – užtikrinti efektyvų, saugų bei patikimą vėjo jėgainės darbą. Todėl iškyla keletą esminių problemų, kurios nagrinėjamos įvairiose mokslinėse, tyrinėjimais grįstose publikacijose. Šio skyriaus tikslas apžvelgti vėjo elektrinių valdymo sistemų pagrindines problemas, kurios pateikiamos įvairiuose straipsniuose tiek pasauliniu, tiek Lietuvos mastu.

Danijos kompanijos „BONUS ENERGY“ publikuotame straipsnyje „The Wind Turbine Components and Operation“ akcentuojama, kad vėjo jėgainių valdymo sistemos susideda iš skirtingų komponentų, kurie atlieka jiems pavestas funkcijas. Šie valdymo sistemos komponentai tarpusavyje yra susieti, tokiu būdu sudarydami vientisą, sudėtingą valdymo sistemą, kuri turi užtikrinti patikimą vėjo elektrinės darbą, užkirsti kelią galimoms susidariusioms pavojingoms situacijoms bei efektyviai kontroliuoti darbą išgaunant kuo didesnę naudingumo koeficientą. Tam tikslui kiekviena valdymo sistema turi nepertraukiamai (nepriklausomai ar jėgainė yra aktyvioje darbo būsenoje) sekti vėjo elektrinės momentinius duomenis bei aplinkos sąlygas. Dauguma sudėtingų valdymo sistemų, realizuotų PLC (Programmable Logic Controller) struktūroje, vėjo jėgainės stebimos būsenos atvaizduojamos personalinio kompiuterio ekrane, naudojant SCADA (Supervisory Control and Data Acquisition) sistemą. Visi parametrai, įvykiai, klaidos ir gedimai kaupiami archyvuose – serveriuose. Valdymo sistema turi ne tik sekti aprašytas būsenas, tačiau tuo pačiu metu valdyti vėjo jėgainės darbą, pagal duotas užduotis arba aplinkos sąlygas. Siekiant užtikrinti patikimą ir saugų visos

vėjo elektrinės darbą, sistema privalo identifikuoti gedimus ne tik vėjo jėgainės konstrukcijoje, tačiau ir pačioje valdymo sistemoje. „The Wind Turbine Components and Operation“ straipsnyje teigiama, kad savikontrolė gali būti vykdoma kelias būdais. Pavyzdžiui žinant generuojamos elektros energijos nominalią reikšmę prie tam tikrų vėjo greičių, galima identifikuoti gedimą. Gedimas įvertinamas, jeigu generuojamos elektros energijos kiekis ryškiai skiriasi nuo nominaliosios vertės (esant tam tikram momentiniam vėjo greičiui). Sekantis būdas aptikti valdymo sistemos gedimus yra dubliavimas, t.y. dubliuojamos valdymo sistemos, lyginamos jų apskaičiuojamos reikšmės, rezultatai bei esant nuokrypoms identifikuojamas gedimas.

Danijos kompanija „BONUS ENERGY“ straipsnyje taip pat nagrinėjama vėjo jėgainės valdymas esant kritiniam vėjui (t.y. kai vėjo greitis siekia $20 \div 25$ m/s). Pateikiama gausybė valdymo algoritmų, kurių funkcija yra ne tik saugiai suvaldyti vėjo jėgainės darbą, tačiau efektyviai išnaudoti vėjo elektrinės galimybes esant tokiems vėjams. Straipsniuose akcentuojama, kad visų tipų vėjo jėgainėms yra būtina valdymo sistema, kuri gebėtų suvaldyti elektrinės darbą pasiekus kritinį vėjo greitį. Taip pat teigiama, kad nesiimant prevencinių priemonių vėjo jėgainės neįmanoma sustabdyti, to pasakoje viršijamas maksimalus galimas generuojamos elektros energijos laikas, konstrukcija gali neatlaikyti apkrovų, generatorius užsiliepsnoti ir pan. Dėl šių priežasčių elektrinės stabdymą būtina aktyvuoti dar nepasiekus paminėto kritinio vėjo greičio, šią sąlyga privalo išpildyti valdymo sistema, kurioje turi būti realizuotas algoritmas gebantis suvaldyti vėjo elektrinės darbą, esant dideliame vėjo greičiui. Žemiau esančiame paveikslėlyje pavaizduota tipinės komercinės vėjo jėgainės darbas pasiekus kritinį vėjo greitį.



1 pav. Vėjo elektrinės darbas pasiekus kritinį vėjo greitį

Šaltinis: „The Wind Turbine Components and Operation“

Pateiktame 1 paveikslėlyje matyti keturios darbo būsenos. Pirmosios būsenos kontroliuojamos valdymo sistemomis su realizuotais skirtingais algoritmais, paskutinė (t.y. viršutinė) darbo būsenos

kreivė iliustruoja vėjo jėgainės darbą be valdymo sistemos. Taigi matyti, kad nesiimant prevencinių valdymo priemonių (artėjant vėjo greičiui prie kritinės reikšmės) rotoriaus sukimosi greitis bei generuojama galia didėja eksponentiškai, kurio pasakoje gali įvykti avarija. Matome, kad labai svarbu, jog valdymo sistema reaguotų pakankamai greitai į aplinkos sąlygas (šiuo atveju vėjo greitį) ir priimtu atitinkamą sprendimą, kurio tikslas užtikrinti vėjo elektrinės saugaus ir efektyvaus darbo optimalų santykį.

Danijos kompanijos „BONUS ENERGY“ straipsnyje „The Wind Turbine Components and Operation“ apibūdinamos vėjo jėgainių valdymo sistemos pagrindinės savybės. Akcentuojama, kad valdymo sistema turi įvykdyti visus valdymo sprendimus atsižvelgiant pagal momentines aplinkos bei vėjo elektrinės būsenas. Tuo pačiu metu sistema turi ne tik vykdyti kontrolę, stebėti momentines reikšmes, tačiau taip pat perduoti duomenys informacijos saugojimo įrenginiams, duomenų archyvavimo tikslo sumetimais. Straipsnyje teigiama, kad vėjo jėgainių valdymo sistemų pagrindas yra mikrovaldiklis, kuris atlieka visus sprendimus bei skaičiavimus pagal EEPROM atmintyje įrašytą valdymo algoritmą.

Danijos kompanijos „BONUS ENERGY“ atstovai siūlo vėjo elektrinę kontroliuoti pagal analogines ir skaitmenines signalų reikšmes. Pagrindiniai analoginiai signalai:

- Įtampos matavimai;
- Srovės matavimai;
- Elektros energijos dažnio matavimai;
- Temperatūros matavimai (generatoriaus, pavarų dėžės, tepalų ir pan.);
- Vėjo greičio matavimas;
- Vėjo krypties nustatymas;
- Rotoriaus sukimosi greičio matavimai;
- Baterijų bloko talpa (aktualu mažoms vėjo jėgainėms iki 10kW).

Taigi valdymo sistema turi gebėti priimti aukščiau pateiktą analoginę informaciją. Tam tikslui naudojami ADC keitikliai (Analogue Digital Converters), kurių pagalba analoginiai signalai konvertuojami į skaitmeninį signalą (apie tai plačiau skaitykite darbo tiriamojoje dalyje). Pagal analoginių signalų reikšmes mikrovaldiklis atlieka vėjo jėgainės kontrolę, bei apskaičiuoja ir vartotojui pateikia papildomą informaciją, tokią kaip elektros energijos galia, energija ir pan.

Pagrindiniai skaitmeniniai signalai:

- Generatoriaus perkaitymas;
- Elektros grandinės nutrūkimas;
- Žingsninių variklių perkaitymas;
- Hidraulinio gedimo signalas;

- Baterijų bloko perkrova / gili iškrova (aktualu mažoms vėjo jėgainėms iki 10kW).

Sekantis vėjo jėgainių valdymo sistemų mokslinės tematikos straipsnis „Coordinated Control Design for Wind Turbine Control Systems” (autoriai: William E. Leithead ir Sergio Dominguez, Strathclyde Uiversitetas, Škotija) teigia, kad norint apibrėžti valdymo sistemų reikalavimus, pirmiausia būtina nustatyti šios sistemos funkcijas. Straipsnio autoriai William Leithead ir Sergio Dominguez pabrėžia, kad pagrindinės vėjo elektrinių valdymo sistemų funkcijos:

- Kontroluoti generuojamą elektros energiją;
- Sumažinti laikinašias apkrovas;
- Užtikrinti sistemos tinkamą dinamiką;
- Padidinti gaminamos elektros energijos kiekį.

Straipsnyje plačiau nagrinėjamos pirmosios dvi valdymo sistemos funkcijos. Taigi, teigiama, kad kontroliuoti generuojamą elektros energijos kiekį galima reguliuojant rotoriaus apsisukimo momentą, t.y. atsižvelgiant į momentinį vėjo greičio dydį, valdymo sistema turi automatiškai nustatyti rotoriaus apsisukimo momento konstantinę vertę, pagal kurią vėjo elektrinė efektyviai generuotų elektros energiją. Šiuo atveju valdymo sistemos uždaviniai: atsižvelgiant į momentinį vėjo greitį, apskaičiuoti rotoriaus apsisukimo momento konstantinį dydį bei atlikti vėjo jėgainės kontrolę pagal šią nustatytąją reikšmę. Norint tai realizuoti straipsnio autoriai siūlo sukurti valdymo sistemai algoritmą pagal kurį rotoriaus apsisukimo momentas būtų valdomas reguliuojant vėjo jėgainės kampą tarp horizontaliosios turbinos ašies ir oro srauto krypties ir/arba keičiant menčių nuolydžio kampą. Tam tikslui pasiekti algoritme būtina naudoti PID (Proportional Integral Derivative) reguliatorius. Taigi tokiu būdu vėjo jėgainės darbas būtų pastovus bei stabilus, nepaisant to, kad vėjo greitis yra pakankamai dažnai kintantis.

Publikuojamame straipsnyje „Modernių vėjo jėgainių savybės ir plėtojimo perspektyvos“ (autorius: Gytis Petrauskas) teigiama, kad vėjo greitis ir kryptis yra nuolat kintantis dydis, todėl atsižvelgiant į vėjo greičio ir krypties pokyčius turbiną turi būti nuolat kontroliuojama valdymo sistemos pagalba. Akcentuojama, kad pagrindinė valdymo sistemos funkcija yra galios optimizavimas, t.y. iš pro turbinos rotoriaus plotą praeinančio oro srauto išgauti galią, artimą varinei turbinos galiai. Sekanti valdymo sistemos funkcija pagal Gytį Petrauską – stabilus turbinos rotoriaus sukimosi greičio palaikymas. Ši funkcija yra naudinga tokio tipo vėjo elektrinėms, kurių generatoriaus apvijos (su generatoriumi per pavarų dėžę mechaniškai sujungtas turbinos rotorius) prie elektros tinklo prijungtos tiesiogiai. Taigi generatoriaus ir jėgainės rotoriaus sukimosi dažnis turi būti suderintas su integruotos energijos sistemos elektros tinklo įtampos dažniu. Jėgainėse, kurių generatoriai prie elektros tinklo jungiami per elektroninį dažnio keitiklį (lygintuvo – inverterio sistema), rotoriaus sukimosi greičio stabilizavimas nereikalingas.

Moderniuose vėjo jėgainėse dažniausiai naudojami šie mechaniniai turbinos galios ir sukimosi greičio valdymo būdai:

- Valdymas keičiant kampą tarp oro srauto krypties ir menčių aerodinaminio profilio ašies;
- Pasyvus valdymas oro sūkuriiais, susidarančiais kapus rotoriaus mentės;
- Kompleksinis valdymas;
- Valdymas keičiant kampą tarp oro srauto krypties ir horizontaliosios turbinos ašies.

Valdymas kampu tarp oro srauto krypties ir menčių aerodinaminio profilio ašies yra pagrįstas kėlimo jėgos priklausomybe nuo šio kampo. Keičiant kampą kintą ir kėlimo jėga, tokiu būdu galima optimizuoti jėgainės generuojamą galią bei rotoriaus sukimosi dažnį. Tai leidžia maksimaliai panaudoti oro srauto energiją, kai jo greitis ne didesnis už vardinį, ir apriboti galią, kai oro srauto greitis padidėja palyginti su projektiniu. Valdymas tarp oro srauto krypties ir menčių aerodinaminio profilio ašies pasižymi tikslumu, be to taikant šį metodą išlaikoma stabili rotoriaus mentės mechaninė apkrova. Pagal straipsnio autorių tam tikslui reikalinga papildoma rotoriaus menčių valdymo sistema.

Naudojant pasyvų valdymą sūkuriiais, rotoriaus mentė nejudamai įtvirtinama tam tikru kampu. Didėjant oro srauto greičiui auga ir kitapus rotoriaus mentės susidarantys oro sūkuriiai, kurie slopina kėlimo jėgą ir tokiu būdu apriboja jėgainės galią. Taip pasiekama, kad jėgainės galia nebūtų didesnė už vardinę.

Kompleksinis valdymas jungia abu anksčiau minėtus būdus. Kol vėjo greitis nesiekia vardinio, naudojamas valdymo kampu, kai vėjo greitis didesnis už vardinį, prasideda galios ribojimas sūkuriiais.

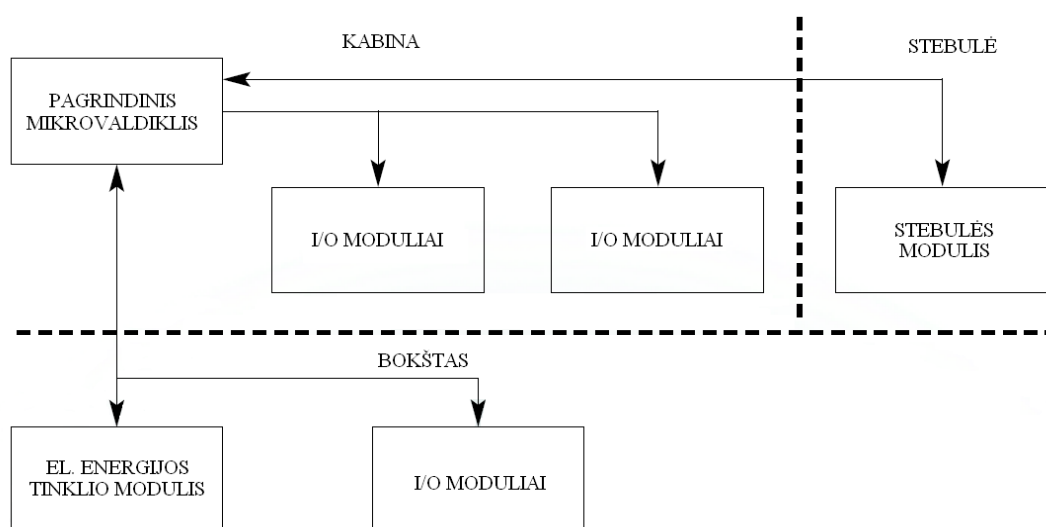
Apžvelgus mokslinių straipsnių (nagrinėjamų Lietuvos ir pasaulio mastu) keliamas problemas bei reikalavimus, galime teigti, kad viena iš pagrindinių keliamų problemų: kaip užtikrinti sistemos patikimą darbą, funkcionalumą, sumažinti galimų gedimų skaičių ne tik vėjo elektrinės konstrukcijos dalyse, tačiau ir pačioje valdymo sistemoje? Taip pat išnagrinėjus ne tik šiame skyriuje pateiktus straipsnius, bet daugybe kitų panašia tematika susijusių, pastebėta, kad aptartos valdymo sistemos susideda iš kelių valdymo komponentų, kuriems priskirtos skirtingos funkcijos. Šie komponentai tarpusavyje sujungti sudaro vientisą kompleksinę ir gana sudėtingą valdymo sistemą. Siekiant sumažinti rizikos laipsnį aparatūros gedimuose, tikslinga būtų atsisakyti kompleksinės valdymo sistemos struktūros, bei pasirinkti kuo paprastesnes, mažiau aparatūrinių dalių turinčias valdymo sistemas. Šiai problemai išspręsti tiktų FPGA (Field Programmable Gate Array) matricos, apie kurias aptarsime sekančiuose skyriuose.

Taip pat straipsniuose pateikta gausybė vėjo jėgainių reikalavimų bei realizavimo būdų. Peržiūrėjus visus reikalavimus galime suformuluoti bendrą teiginį, kuris aprėpia valdymo sistemų reikalavimų spektrą bei paskirtį: vėjo elektrinių valdymo sistema turi užtikrinti patikimą vėjo elektrinės darbą, užkirsti kelią galimai susidariusioms pavojingoms situacijoms bei efektyviai kontroliuoti darbą išgaunant kuo didesnę naudingumo koeficientą.

1.3 Esamų vėjo jėginių valdymo sistemų rinkos apžvalga

Siekiant pagerinti gamtosaugines sąlygas, vis plačiau naudojama vėjo energija. Šiuolaikinėse jėgainėse vėjo energija verčiama į elektros energiją, kuri naudojama buityje, o perteklius atiduodamas į tinklą. Didėjantis skaičius šalių diegiančių vėjo jėgaines, rodo gebėjimą pritaikyti inovatyvias technologijas naudojant atsinaujinančius energijos išteklius (AEI). Klimato pasikeitimo problemos, aukštos naftos kainos skatina įstatymų leidybą susijusią su alternatyviają energetika, todėl atsinaujinančios energijos rinka vis dar auga. Sparčiai tobulėjant vėjo energetikos technologijoms, iškyla poreikis vystyti ir tobulinti valdymo sistemas, kurios užtikrintų saugų bei patikimą vėjo elektrinių eksploatavimą. Todėl pasaulyje yra daugybė įmonių, kurios vysto šia nišą bandydamos padengti vis didesnius vėjo elektrinių keliamus reikalavimus.

Šiuo metu plačiausiai paplitusios standartinės vėjo jėginių sistemos, kurios realizuotos PLC pramoninių valdiklių pagrindu. Vieni iš pagrindinių vėjo jėginių bei joms skirtų pramoninių valdiklių gamintojai yra danai. Jų rinką sudaro 35 šalys, iš kurių didžiausios yra Vokietija, Indija, Olandija, Kinija, Ispanija, Didžioji Britanija ir Švedija. Viena iš Danijos kompanijų, kuri gamina valdymo sistemas vėjo elektrinėms yra „kk – Electronics a/s“. Ši kompanija siūlo modulinės konstrukcijos pramoninius vėjo elektrinės valdiklius su SCADA sistema, kurie paplitę ne tik Danijos regione, tačiau ir visame pasaulyje. Šie valdikliai skirti horizontalaus tipo komercinėms vėjo jėgainėms valdyti. Kompanijos „kk – Electronics“ siūloma sistema sudaryta iš kelių komponentų – modulių (tinklo moduliai, įėjimų/išėjimų moduliai ir t.t.). Vartotojams yra siūloma bazinė valdymo sistemos konstrukcija, tačiau esant didesniam vartotojo poreikiui – siūloma papildomai išplėsti sistemos galimybes, išnaudojant modulinės konstrukcijos savybes. Žemiau esančiame paveikslėlyje patekta Danijos įmonės „kk – Electronics“ valdiklio struktūra:



2 pav. Vėjo elektrinės valdymo sistemos struktūra

Šaltinis: „The Wind Turbine Components and Operation“

Paveikslėlyje matyti, kad sistema sudaryta iš pagrindinio mikrovaldiklio, kelių modulių, skirtų tam tikrai vėjo elektrinės daliai kontroliuoti bei jų momentinėms būsenoms stebėti. Išnagrinėjus šios vėjo jėgainių valdymo sistemos technines specifikacijas, galime teigti, kad sistema sudaryta iš vieno pagrindinio mikrovaldiklio bei eilę šalutinių mikrovaldiklių, kurie integruoti skirtinguose modeliuose (I/O, tinklo, stebulės ir pan.). Taigi ši valdymo sistema susideda iš:

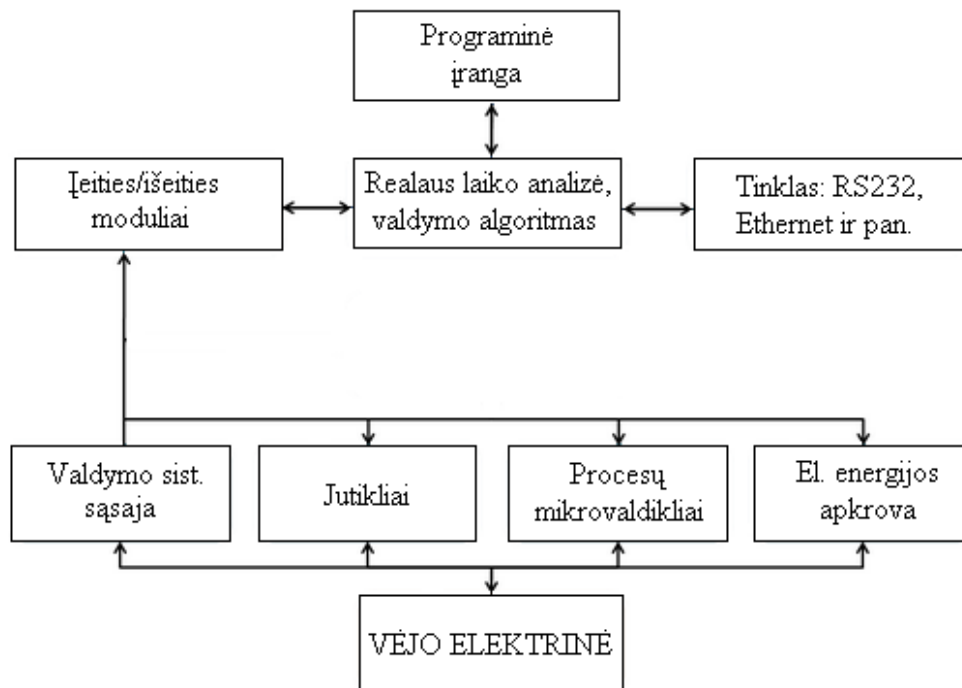
- **Pagrindinio mikrovaldiklio:** Intel 386, 32-bitų; sparčioji atmintis 8 MB; RAM operatyvioji atmintis 2 MB; su 10/100 Mb „Ethernet“ jungtimi; 4 magistralės su CAN protokolu; su 3 skaitmeniniais išėjimo signalais; 2 reliniais išėjimais (NC/NO – Normal Close/Open); 2 skaitmeniniais įėjimo signalais; 1 analoginiu įėjimo signalu (temperatūros matavimui). Šis mikrovaldiklis turi funkciją save testuoti ir aptikti gedimus.
- **El. energijos tinklo modulis:** Texas Instruments kompanijos DSP (Digital Signal Processor) procesorius; su 3 fazių įtampos/srovės įėjimais; tiristoriniu išėjimu (3 fazių kontrolei); su 4 x PT100 temperatūriniais įėjimais; 4 skaitmeniniais įėjimo/išėjimo signalais bei reliniais išėjimais (NC/NO – Normal Close/Open); magistralė su CAN protokolu. Šis procesorius turi funkciją save testuoti ir aptikti gedimus.
- **Stebulės modulis:** Texas Instruments kompanijos DSP (Digital Signal Processor) procesorius; su 10 reliniais išėjimais (NO – Normal Open); 7 skaitmeniniais įėjimo signalais; 4 analoginiais įėjimų signalais (0...20mA); 3 analoginiais įėjimų signalais (0...10V); su 1 x PT100 temperatūriniu įėjimu; 6 PWM (Pulse Width Modulation) išėjimais; su RS485 fiziniu ryšiu; magistralė su CAN protokolu. Šis procesorius turi funkciją save testuoti ir aptikti gedimus.
- **I/O modulių** – Atmel AVR RISC procesorius; su 10 x PT100 temperatūriniais įėjimais; su 4 analoginiais įėjimų signalais; 24 skaitmeniniais įėjimo signalais; 1 analoginiu išėjimo signalu; su 14 reliniais išėjimais (NO – Normal Open); 4 skaitmeniniais išėjimo signalais; su RS485 fiziniu ryšiu; magistralė su CAN protokolu. Šis procesorius turi funkciją save testuoti ir aptikti gedimus.

Pagrindinio mikrovaldiklio užduotis – optimizuoti komponentų operacijas, atlikti globalias valdymo komandas, sekti sistemos būseną bei identifikuoti gedimus. Kiekvienas iš valdymo sistemos komponentų turi atskiras užduotys, pavyzdžiui stebulės modulis, skirtas kontroliuoti vėjo jėgainės menčių kampui (angl. „pitch control“). Elektros energijos tinklo modulis, skirtas kontroliuoti bei stebėti generuojamą elektros energiją. I/O (įėjimas/išėjimas) moduliai, skirti priimti įvairaus tipo informaciją (iš daviklių), ją perduoti pagrindiniam mikrovaldikliui ir pagal mikrovaldiklio valdymo algoritmą išvesti valdymo signalą tam tikrai vėjo elektrinės daliai. Taigi matyti, kad ši valdymo sistema gana sudėtinga, nes susideda iš kelių komponentų su skirtingais mikrovaldikliais. Šie

mikrovaldikliai sujungti tarpusavyje magistralėmis (naudojant CAN protokolą informacijai priimti/perduoti) sudaro kompleksinę vėjo elektrinės valdymo sistemą. Danų kompanijos „kk – Electronics“ vėjo jėginių valdymo sistema turi galimybę turėti ryšį su PC (personaliniu kompiuteriu), tokiu būdu SCADA vizualizacijos sistemoje vartotojui pateikiama pilna informacija apie vėjo elektrinės darbą bei suteikiama galimybė valdyti tam tikras vėjo jėgainės būsenas. Valdymo, momentinių reikšmių bei gedimų informacija saugojama serveriuose.

Amerikiečių kompanija „National Instruments“ (toliau NI), kuri turi filialus visame pasaulyje taip pat turi pasiūlytą vėjo jėginių valdymo sistemas. Labiausiai paplitusios šios kompanijos valdikliai yra „Compact RIO Advisor“ ir „PXI Advisor“ modulinės struktūros valdikliai. „Compact RIO Advisor“ valdikliai yra kompaktiški, mažesnių galimybių valdikliai, tačiau jie itin tinkantys atlikti diagnostikai arba pritaikyti mažesnio galingumo vėjo jėgainėms. „Compact RIO Advisor“ valdiklis turi nuo 4 iki 32 I/O (įėjties/išeities) kanalų, USB arba Ethernet jungtis, nešiojamą arba fiksuotą valdymo sistemos konstrukciją.

„PXI Advisor“ valdymo sistema turi daug platesnes galimybes bei aukštesnes charakteristikas, todėl šis valdiklis tinkamas sudėtingoms vėjo elektrinėms. „PXI Advisor“ valdiklis turi virš 500 I/O (įėjties/išeities) kanalų, Ethernet, RS232 (Modbus protokolas) jungtis bei žymiai didesnę greitaveiką. Visi šie „National Instruments“ įtaisai, sudaryti iš kelių komponentų (su mikrovaldikliais), kurie atlieka jiems pavestas funkcijas. NI kompanijos valdymų sistemų blokinė schema pateikta 3 paveikslėlyje:



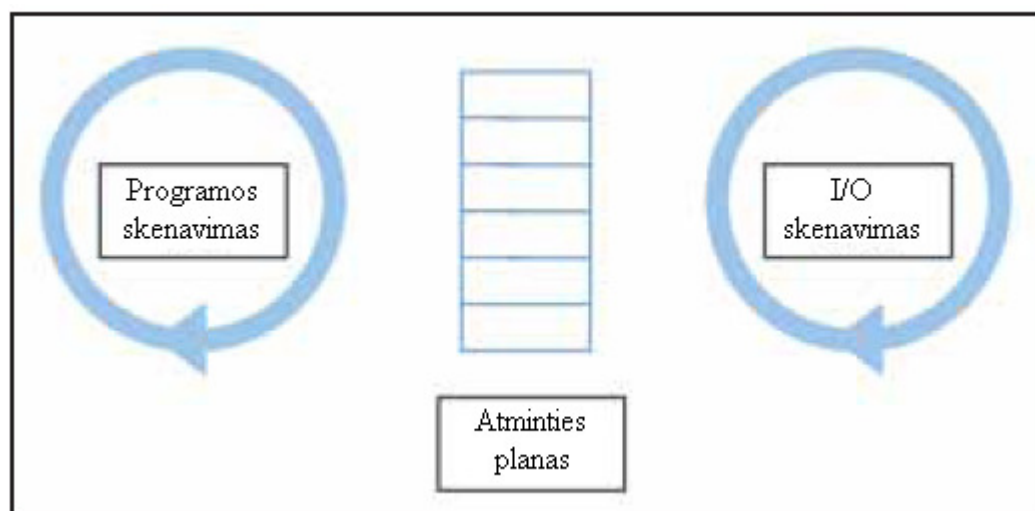
3 pav. Vėjo elektrinės valdymo sistemos blokinė schema

Šaltinis: sudaryta autoriaus

NI sistema turi galimybę stebėti vėjo jėgainės būsenas: temperatūrą, vibraciją, generuojamos elektros energijos, hidraulinių tepalų momentines reikšmes. Taip pat sistema turi galimybę valdyti vėjo jėgainę pagal momentines būsenas arba vartotojo komandas. Esant poreikiui išplėsti sistemos galimybes, vartotojui yra pateikiama gausybė modulinės konstrukcijos modulių, kurie jungiami į bendrą sistemą. Visa informacija pateikiama bei būtinos valdymo funkcijos vykdomos programinėje įrangoje HMI (Human Machine Interface).

„National Instruments“ įmonė kuria ne tik standartinius valdiklius vėjo jėgainėms, tačiau taip pat bando sukurti inovacines technologijas, naudojant FPGA matricas. Taigi vėjo jėgainių valdymo sistemų rinkoje galima rasti NI kompanijos sukurtus modulius, kurie sudaryti iš FPGA architektūrų. Ši modulį galima integruoti į standartini „CompactRIO“ valdiklį. Akcentuojama, kad sudėtingėjant technologijoms, standartinės valdymo sistemos tampa nebepajėgios aprėpti visas būtinas funkcijas bei reikalavimus, todėl naudojama kombinuota valdymo sistema, kurioje lygiagrečiai dirba FPGA ir PLC technologijos.

Tipinis PLC valdiklio veikimo principas yra standartizuotas IEC-61131-3 standartu. Šis standartas parengtas tarptautinės elektrotechnikos komisijos IEC (International Electrotechnical Commission), kuriame detalizuotas sistemos vykdymo procedūros, programavimo kalbos ir pan. Todėl visi rinkoje esantys PLC valdikliai (taip pat PLC skirti vėjo jėgainėms) turi vienodą programos veikimo algoritmą:



4 pav. PLC valdymo sistemos standartizuota veikimo architektūra

Šaltinis: „Combining PLC and FPGA architectures“

Trumpai pateiksime šios PLC standartizuotos veikimo architektūros principus. Tarkime norint, kad valdymo sistema kas sekundę junginėtų lemputę, PLC turi pasitelkti trejetą programos komponentų (funkcijų): programos logikos skenavimas, atminties planas ir I/O (įvesties/išvesties) skenavimas. Pirmiausia programos logikoje reikia įrašyti reikšmę „TRUE“ kintamajam (pvz. kurio

pavadinimas LEDValue). Šis kintamojo pavadinimas patalpinamas atminties plane. Tada I/O skenavimas yra vykdomas, kuris priskiria fizinei valdiklio šynai (kanalui) kintamojo LEDValue (parinkto iš atminties plano) reikšmę. Taigi ši programos valdymo architektūra yra tinkama nuosekliems procesams vykdyti, tačiau sudėtingėjant sistemoms, kyla poreikis procesus atlikti ne tik nuosekliai, tačiau ir lygiagrečiai. Todėl NI tarptautinė kompanija savo valdymo sistemai yra numačiusi FPGA modulius, kurie gali būti integruojami į PLC valdiklio modulinę konstrukciją.

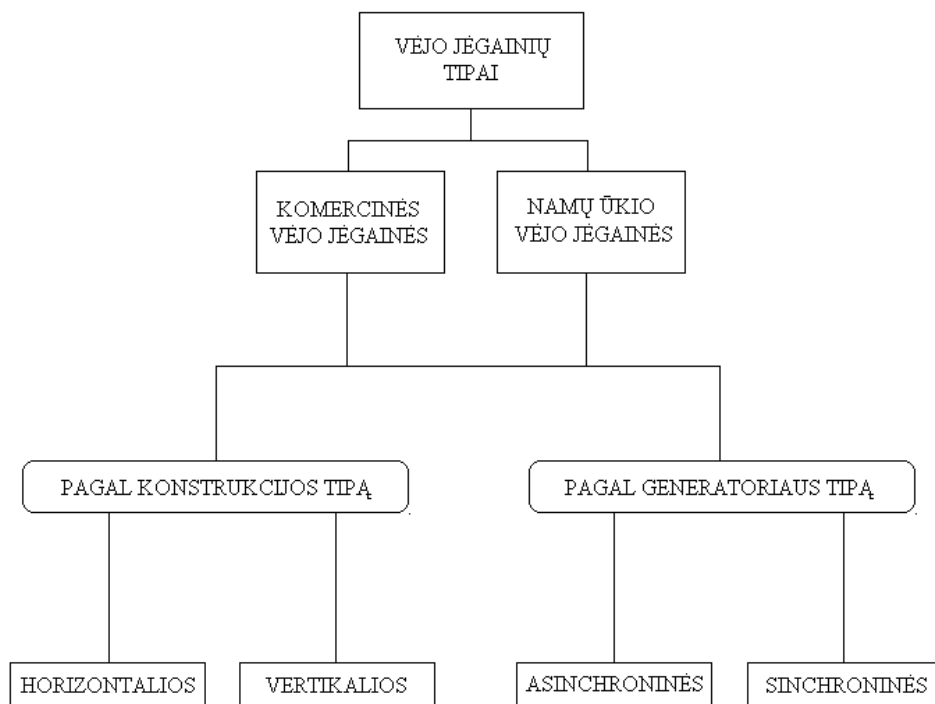
Sudėtingiems valdymo sistemos algoritmams paranku naudoti FPGA (Field Programmable Gate Array) architektūrą, ne tik todėl, kad procesai gali būti vykdomi lygiagrečiai, žymiai didesnė greitaveika, tačiau ir todėl, kad tokia valdymo sistema gali būti pilnai suprogramuota ir ištestuota dar neintegruojant į bendrą valdymo sistemą. FPGA matricos yra sudarytos iš daugybės loginių ventilių (pvz. Spartan3E turi 500 000 ventilių). Paprastai tariant, mašininės programavimo kalbos pagalba loginiai ventiliai yra sujungiami tarpusavyje, tokiu būdu sudarydami loginius elementus, kurie funkcionuoja pagal programuotojo parašytą valdymo algoritmą. FPGA matricos dažnai yra naudojamos tada, kai mikrovaldikliai (pvz. RISC tipo) nebepajėgūs įvykdyti valdymo algoritmo.

Vėjo jėgainių valdymo sistemų rinkoje šiuo metu retai naudojami valdikliai, kurie dirba remiantis vien tik FPGA matricos architektūra, tačiau tarptautinė amerikiečių kompanija „Acromag“ siūlo šį nestandartinį sprendimą, t.y. vėjo jėgainių valdymo sistemas „Acromag PMC FPGA“, kurios realizuotos FPGA matricomis. Įmonė pristatydamą šį valdiklį taip pat akcentuoja, kad valdymo sistemos (su FPGA matrica) greitaveika yra daug didesnė (procesai vykdomi lygiagrečiai) nei įprastų standartinių valdiklių, platus I/O (įeičių/išeičių) diapazonas bei pasiekiamas žymiai didesnis lankstumas nei PLC valdiklių skirtų vėjo elektrinėms. „Acromag PMC FPGA“, sistemos valdymo algoritmas realizuojamas mašinine programavimo kalba (VHDL, Verilog, SystemC).

1.4 Projektuojamos sistemos sprendimų būdų analizė

Vėjo jėgainės yra skirstomos pagal įvairius kriterijus, pavyzdžiui pagal generatoriaus tipą (sinchroniniai ar asinchroniniai), pagal konstrukciją (horizontalios arba vertikalios vėjo jėgainės) arba pagal paskirtį (komercinės arba namų ūkio). Dažniausiai komercinės vėjo jėgainės yra didelio galingumo (nuo 10 KW iki 2 MW arba daugiau). Tokio tipo jėgainės dažniausiai būna prijungtos prie elektros energijos tinklo, taigi visą generuojamą elektros energiją yra perduodama tiesiogiai į tinklą. Namų ūkio vėjo jėgainėms – mažo galingumo (iki 10 KW), žymiai paprastesnės konstrukcijos elektrinės. Dažniausiai tokio tipo jėgainės sujungtos su lokaliu namų ūkio elektros energijos tinklu. Esant poreikiui vartotojas naudoja generuojamą elektros energiją (esant vėjui), tačiau jeigu poreikio nėra, energija yra akumuliuojama baterijų blokuose, tokiu būdu elektra eikvojama tada, kai vartotojui atsiranda poreikis. Paskutiniu metu, vystantis elektromobilių technologijomis, kalbama, kad mažo

galingumo vėjo elektrinės gali būti pritaikomos automobilių akumuliatoriams pakrauti. Žemiau esančiame 5 paveikslėlyje matyti vėjo jėgainių klasifikavimo struktūra:

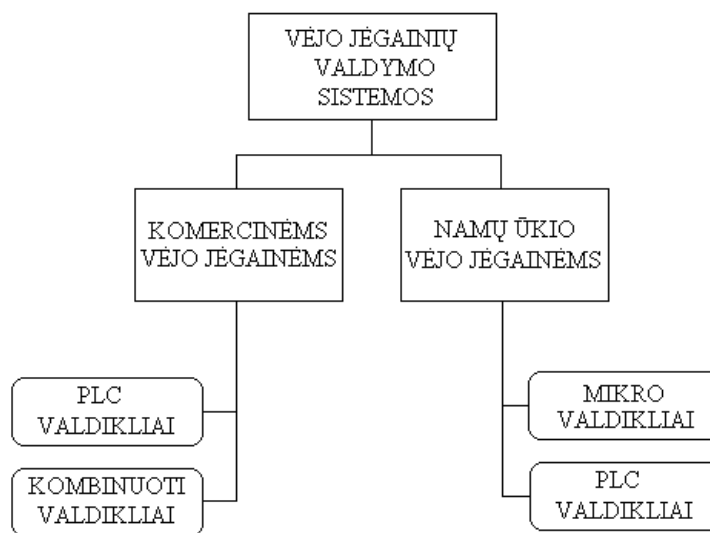


5 pav. Vėjo jėgainių klasifikavimo struktūra

Šaltinis: sudaryta autoriaus

Taigi pagal vėjo jėgainės tipą turi būti parenkama valdymo sistema, kuriai keliami skirtingi reikalavimai. Didelio galingumo (komercinės paskirties) vėjo elektrinių valdymo sistemoms keliami itin aukšti reikalavimai bei valdymo algoritmai priskirti skirtingiems sistemos komponentams. Ankstesniuose skyriuose akcentuota, kad tokios sistemos kiekvienas komponentas atlieka skirtingas valdymo funkcijas (mikrovaldiklių pagalba), pavyzdžiui: generuojamos elektros energijos grandinės kontrolė, menčių nuolydžio kampo reguliavimas, vėjo jėgainės kampo tarp horizontaliosios turbinos ašies ir oro srauto krypties reguliavimas bei apsaugą nuo kritinių vėjų. Todėl iškyla viena iš pagrindinių problemų – nepatikimumas. Valdymo sistemai (sudarytai iš daugybės komponentų, kuriems priskirtos skirtingos funkcijos) iškyla didesnė galimų gedimų grėsmė. Kadangi komercinių vėjo elektrinių darbas turi būti kontroliuojamas nenutrūkstamai, todėl būtina išvengti arba bent identifikuoti valdymo sistemos gedimus. Tam tikslui sistema turi atlikti savikontrolę arba naudoti dubliuotą valdymo sistemos technologiją. Sekantis būdas išvengti kompleksinių valdymo sistemos komponentų – FPGA matricos. Valdikliai, kurių struktūra paremta FPGA matricomis, pasižymi ne tik didele greitimeika, bet sumažina „daugiaprocesorinę“ struktūrą. Taigi FPGA matricose realizuoti skirtingų funkcijų valdymo algoritmai (procesai) gali būti vykdomi lygiagrečiai, todėl tokioje sistemoje nebereikia naudoti kompleksinės valdymo sistemos struktūros.

Mažo galingumo (namų ūkio) vėjo jėgainėms valdymo sistemų reikalavimai yra taikomi žymiai paprastesni, nes siauresnis funkcionalumo diapazonas. Mažesnio galingumo vėjo jėgainių valdymo sistemai pakanka vieno mikrovaldiklio arba nesudėtingo PLC valdiklio, kurie užtikrintų vėjo elektrinės kontrolę, todėl sistemos gedimų identifikavimas nėra būtina, tačiau rekomenduojama sąlyga. Namų ūkio vėjo jėgainių valdymo sistemos vienos iš pagrindinių funkcijų: generuojamos elektros energijos kontrolė, apsauga nuo kritinių vėjų bei baterijų bloko įkrovos/iškrovos kontrolė.



6 pav. Vėjo jėgainių valdymo sistemų klasifikavimo struktūra

Šaltinis: sudaryta autoriaus

Aukščiau esamame skyriuje (1 dalies 1.1 skyrius) nagrinėjamos problemoms išspręsti, greitaveikai, saugumui ir funkcionalumui padidinti galima naudoti sprendimo būdą su FPGA matricomis. Siekiant pamatyti skirtumą tarp įprastos kompleksinės vėjo jėgainių valdymo sistemos (naudojant MCU mikrovaldiklius) ir sistemos, kurios architektūra paremta FPGA matricomis, aprašysime FPGA ir MCU (Micro Controller Unit) struktūrą bei veikimo principą. Taip pat atliksime vėjo jėgainių valdymo sistemos realizacijas naudojant FPGA ir MCU technologijas. Abiem realizacijom atlikti, sukursime vienodus valdymo algoritmus bei atliksime tyrimą esant vienodam 4 MHz sinchroninio signalo dažniui. Taigi, darbo *užduotis*:

- sukurti dvi realizacijas (FPGA ir MCU pagrindu) bei jas iširti, naudojant tą patį valdymo algoritmą ir dažnį;
- tarpusavyje palyginti gautus modeliavimo rezultatus;
- pateikti vėjo jėgainių valdymo sistemų realizacijų (FPGA ir MCU pagrindu) skirtumus;
- Išsiaiškinti ar FPGA architektūrą galima panaudoti vėjo energetikoje, taip išvengiant išskylančias problemas;
- Įvertinti kokio tipo vėjo jėgainėms FPGA architektūra būtų tinkamiausia.

1.5 FPGA matricos ir MCU mikrovaldiklių architektūrų apžvalgos

1.5.1 FPGA matricos architektūros apžvalga

FPGA (Field Programmable Gate Array) yra programuojamos puslaidininkinės schemos, kurios sudarytos iš konfigūruojamų loginių blokų CLB (Configurable Logic Blocks) matricos, kai CLB konfigūruojami loginiai blokai sujungti programuojamomis jungtimis. Lyginant FPGA matricas su ASIC (Application Specific Integrated Circuits) technologija, galime teigti, kad ASIC įtaisai kuriami konkrečiam projektui, be galimybės integruoti papildomas (projekte nenumatytas) funkcijas, o FPGA matrica gali būti programuojama pagal įvairių projektų reikalavimus, neįnešant esminių aparatūrinių pakeitimų.

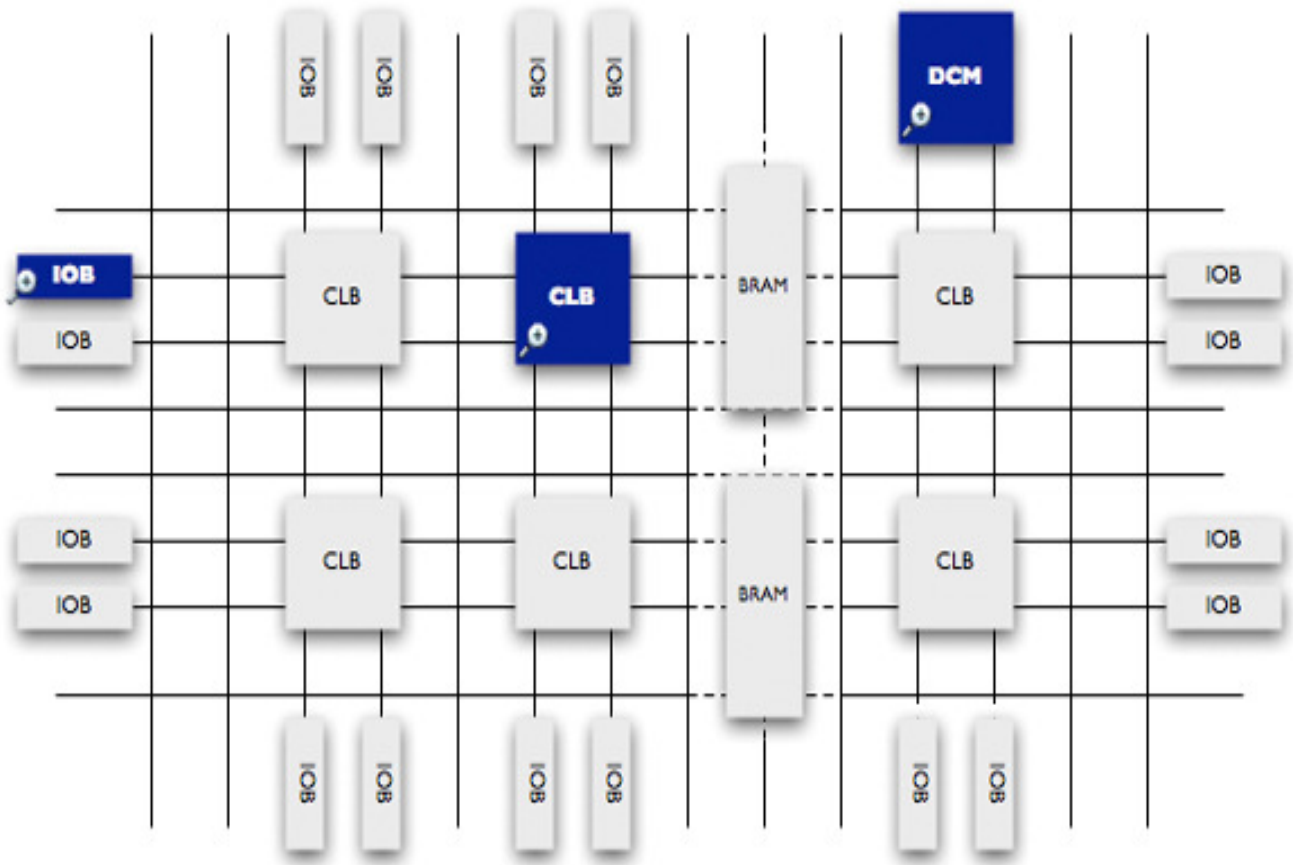
FPGA matricos įvairiai skirstomos, tačiau mes išskirsime du tipus, tai OPT (One Time Programmable) tipo, kurios gali būti tik vieną kartą programuojamos arba SRAM tipo – daugybę kartų konfigūruojamos (perprogramuojamos) technologijos. SRAM sistemos yra daug plačiau paplitusios nei OPT, todėl šio skyrelio dalyje plačiau apžvelgsime šios sistemos struktūrą. Siekiant apibrėžti FPGA matricų SRAM tipo technologijos pagrindinius privalumus, galime teigti, kad nauji projekto pakeitimai gali būti greitai įgyvendinami ir testuojami tol, kol gaunamas norimas rezultatas. Be to FPGA matricos gali būti suprogramuotos savikontrolei atlikti, kai valdymo sistemai paduodamas maitinimas. Aptikus netinkamus pokyčius, matricos gali būti perprogramuojamos, siekiant įvykdyti sistemai iškeltus tikslus. Dar vienas svarbus privalumas yra tai, kad FPGA struktūroje galima lengvai aptikti ir identifikuoti problemas atsiradusias šios technologijos struktūroje. Taipogi galime pabrėžti, kad FPGA matricų SRAM elementas sukurtas, naudojant tas pačias CMOS (Complementary metal–oxide–semiconductor) technologijas, kaip ir visoms kitoms FPGA architektūros dalims, taigi nereikia naudoti jokių papildomų žingsnių ar būdų šiems komponentams sukurti.

Kiekvienas gamintojas turi skirtingą FPGA architektūrą, tačiau pagrindiniai elementai ir struktūra išlaikomi vienodi. Tarkime Xilinx kompanijos FPGA matricų architektūra susideda iš šių pagrindinių blokų:

- konfigūruojamų loginių blokų CLB (Configurable Logic Blocks);
- konfigūruojamų I/O blokų;
- skaitmeninio sinchroninio signalo valdymo bloko DCM (Digital Clock Management);
- programuojamų vidinių jungčių (Interconnections).

Visų viršuje išvardintų architektūros dalių skaičius ir tipas įtakoja FPGA matricos galimybes, todėl įvairių gamintojų FPGA matricos yra klasifikuojamos pagal architektūros sudedamųjų dalių kiekį. Kiekviena sudedamoji architektūros dalis atlieka tam tikrą funkciją, pavyzdžiui CLB blokas - viena iš pagrindinių dalių, kuri sudaryta iš loginių dalelių (celių). Šių dalelių pagalba formuojamos

FPGA matricos vykdymo funkcijos. I/O blokas – skirtas vidiniams FPGA blokams sujungti su išoriniais elementais. Programuojamos jungtys naudojamos vidinių elementų tarpusavio ryšiui sudaryti, o DCM blokas – skirtas taktinio dažnio konfigūravimui. Toliau šiame poskyryje plačiau aptarsime šiuos FPGA architektūros elementus.



7 pav. FPGA matricos blokinė struktūra
Šaltinis: XILINX techninė dokumentacija

Paveikslėlyje matyti pagrindinius XILINX kompanijos FPGA matricos (pvz. Spartan šeimos) architektūros sudedamąsias dalis. Skirtingi gamintojai architektūros blokus vadina skirtingais vardais.

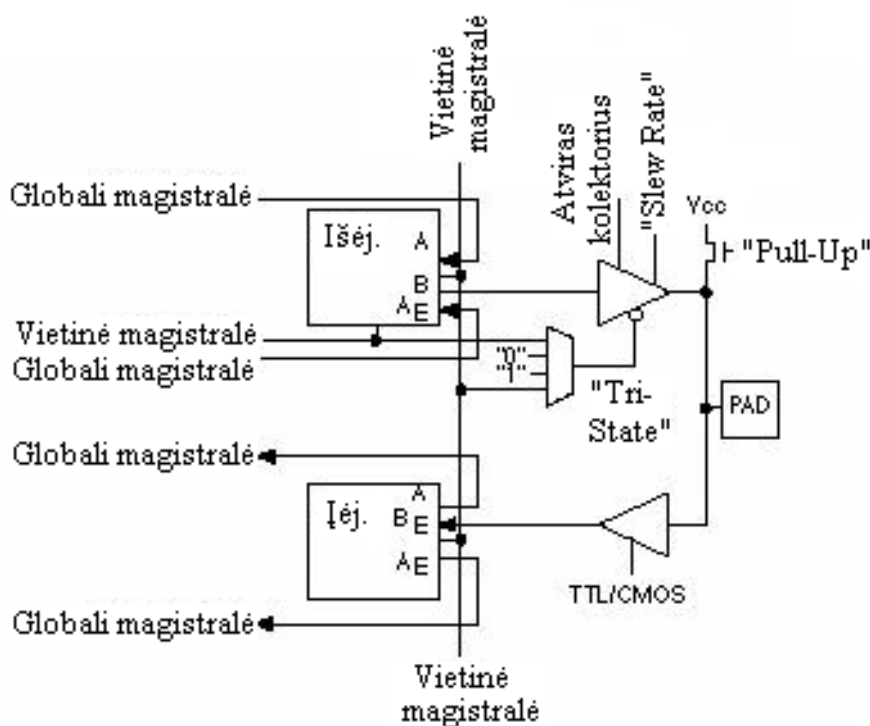
Žemiau pateikiame pavyzdžius gamintojų bei jų FPGA matricos SRAM tipo šeimų variantus:

- Altera FLEX šeima;
- Atmel AT6000 ir AT40K šeima;
- Lucant Technologies ORCA šeima;
- XILINX XC4000 ir VIRTEX šeima.

Turėdami daugiausiai patirties su XILINX kompanijos gaminiais, todėl darbo tyrimui atlikti naudojome SPARTAN šeimos FPGA matricą. Kadangi nagrinėsime šios kompanijos FPGA architektūrą todėl naudosime jų terminologiją. Taigi, pateksime kiekvieno FPGA architektūroje esančio bloko paskirtį bei aprašymą.

IOB – konfigūruojamas I/O (įėjties/išėjties) blokas. Šie blokai naudojami signalų perdavimui/išvedimui į/iš konfigūruojamų loginių blokų (CLB). Kitais žodžiais tariant I/O blokai skirti sudaryti ryšį tarp vidinių FPGA matricos signalų su išoriniu I/O (įėjimų/išėjimų) kanalu. IOB blokai sudaryti iš įėjimų/išėjimų (trijų būsenų) buferių ir atviro kolektoriaus valdymu. Taip pat paprastai naudojami „pull-up“ ir retkarčiais „pull-down“ tipo rezistoriai, „Flip-Flop“ trigeriai, kurių pagalba sinchroninis signalas perduodamas tiesiogiai į šynas išvengiant didelio signalo vėlinimo.

Aukšto ir žemo signalų reikšmės aktyvuojamos keičiant signalo poliškumą. Nustatyti signalo aukšto/žemo frontų kilimo/nusileidimo laiką galima programuojant signalo poliškumo pasisukimo uždelsimą (Slew Rate). Žemiau esančiame paveikslėlyje matyti IOB sudedamieji elementai bei jungtys tarp jų (žiūr. 8 paveikslėlį):



8 pav. FPGA matricos konfigūruojamas I/O blokas

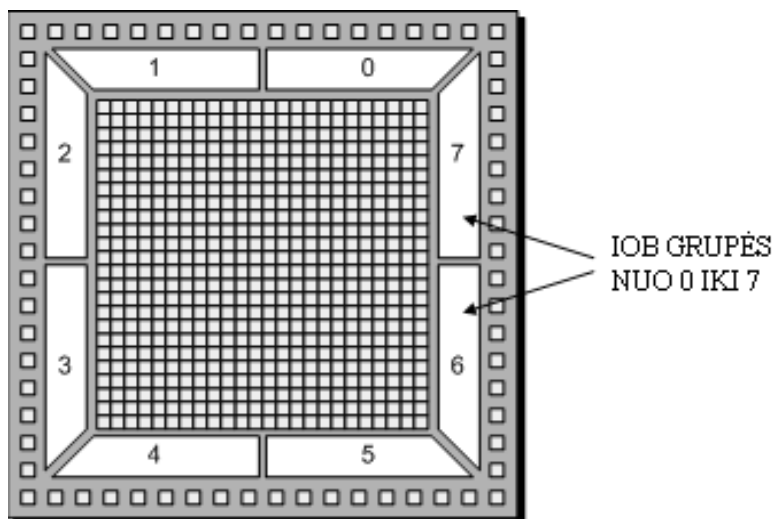
Šaltinis: „Introduction to CPLD and FPGA Design“, autorius Bob Zeidman

Šiuo metu FPGA matricos palaiko daugybę I/O (įėjties/išėjties) standartų, kurie suteikia įvairiapusišką sąsają su skirtingos paskirties sistemomis (tame tarpe ir vėjo jėgainėmis). Šiuo atveju „standartas“ apibūdina elektrinių signalų charakteristikas, pavyzdžiui standartuose aprašomi loginio „1“ ir „0“ įtampos lygiai ir pan. Taigi tokių standartų yra sukurta gausybė, todėl būtų sudėtinga ir nepraktiška sukurti FPGA matricas, kurios palaikytų įvairius standartus. Tam tikslui I/O (įėjties/išėjties) gali būti konfigūruojamos (programuojamos) pagal poreikius pasirenkant tam tikrą standartą. Todėl FPGA matricoje I/O grupuojami atskiruose grupėse su tam tikru kiekiu programuojamų įėjčių/išėjčių (žiūr. 9 paveikslėlį). Tokiu būdu suteikiama galimybė IOB blokams palaikyti skirtingus I/O standartus,

tai suteikia pakankamai didelį FPGA matricų lankstumą. Taigi, paveikslėlyje pavaizduotuos IOB grupės (nuo 0 iki 7) gali būti programuojamos skirtingiems I/O elektrinių signalų standartams.

Galime išskirti pagrindines FPGA matricos IOB blokų savybes:

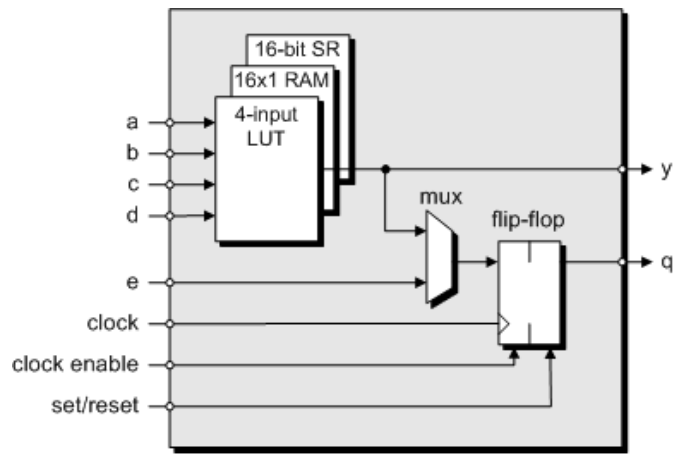
- Gali būti programuojami, kaip įėjimai arba išėjimai;
- Apsauga nuo elektrostatinės iškrovos ESD (Electrostatic discharge protection);
- Išėjimo signalo charakteristikų konfigūracija (greitis, signalo fronto uždelsimas ir pan.);
- Įvairių standartų palaikymas (PCI, LVCMOS, LVDS ir t.t.), programuojant IOB grupes.



9 pav. FPGA matricos konfigūruojamas I/O blokas

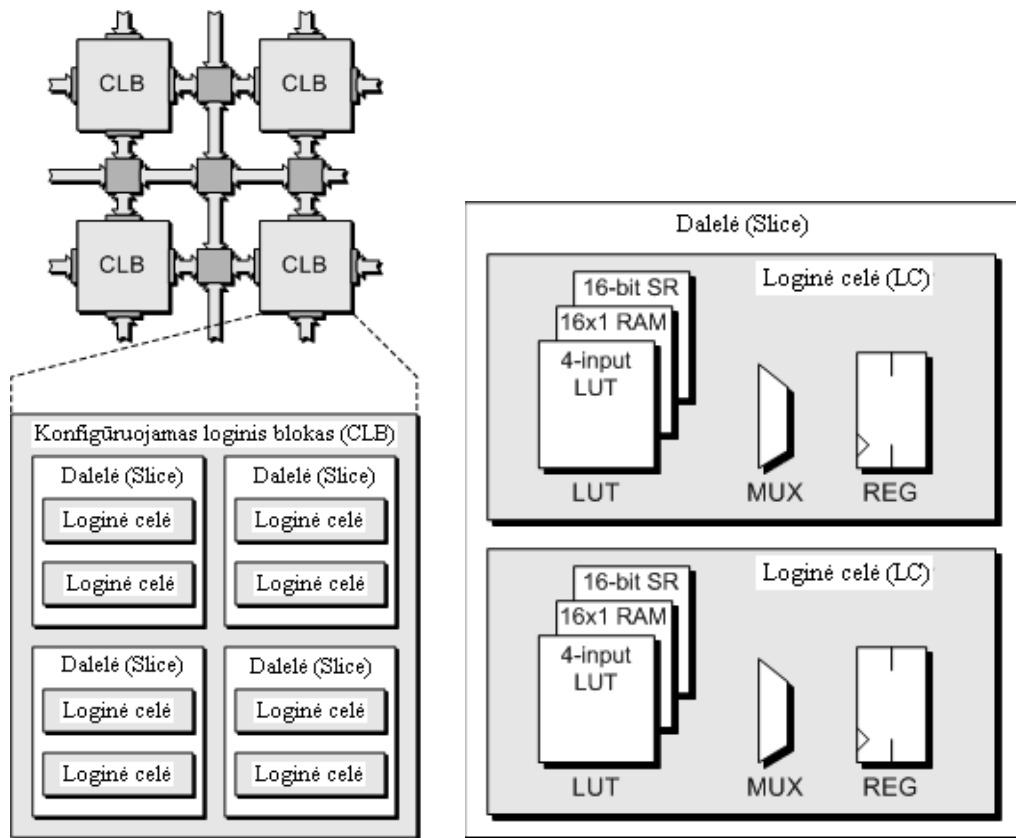
Šaltinis: „FPGA Architectures form A to Z“ straipsnis parengtas EE Times Design

CLB (Configurable Logic Blocks) – konfigūruojamas loginis blokas. Tai pagrindinis FPGA blokas, kurio kiekvienas elementas sujungtas su kelio paskirstymo elementu (angl. Switch Matrix). Šis atitinkamai prijungtas su pagrindiniu kelio paskirstymo elementu. Priklausomai nuo gamintojo, CLB elementas gali susidėti iš skirtingos poros loginių dalelių (angliškai vadinama „Slices“). Pavyzdžiui 11 paveikslėlyje matyti, kad Xilinx kompanijos FPGA matricų CLB blokai gali turėti porą arba dvi poras loginių dalelių (angl. „Slices“). Toliau nagrinėjant konfigūruojamų blokų CLB architektūrą, pastebėtume, kad kiekviena loginė dalelė susideda iš dviejų loginių celių, kuriose loginių ventilių junginiai sudaro tokius elementus kaip: paskirstytuvus (MUX), registrus, 4 įėjimų loginių funkcijų lenteles (LUT), kurios gali dirbti kaip 16x1 RAM arba 6 bitų SR poslinkio registras. CLB bloko esančios loginės dalelės (angl. „Slice“) celės (angl. „Logic Cells LC“) elementams naudojamas bendras sinchroninio dažnio (angl. „clock“, „clock enable), perkrovimo (angl. reset) signalai operacijoms atlikti. Kiekvienas iš išvardintų CLB bloko architektūros sudedamųjų dalių yra sujungtas vidiniais sujungimais (žiūr. 10 pav.). Greičiausiai signalas sklinda tarp loginės celės esančių elementų, o lėčiausiai tarp konfigūruojamų CLB blokų.



10 pav. FPGA matricos konfigūruojamo CLB bloko dalelės loginės celės (LC) sandara
 Šaltinis: „FPGA Architectures form A to Z“ straipsnis parengtas EE Times Design

Siekiant išlaikyti aiškumą 10 paveikslėlis pateiktas supaprastinta forma. Matyti, kad loginės celės registras gali būti sukonfigūruotas (suprogramuotas) kaip „Flip – Flop“ arba „Latch“ tipo trigeris. CLB bloko loginė celė be išvardintų elementų (MUX, LUT ar registrų) gali turėti keletą papildomų loginių elementų aritmetinėms funkcijoms atlikti.



11 pav. FPGA matricos konfigūruojamas CLB blokas is sudedamosios dalys
 Šaltinis: „FPGA Architectures form A to Z“ straipsnis parengtas EE Times Design

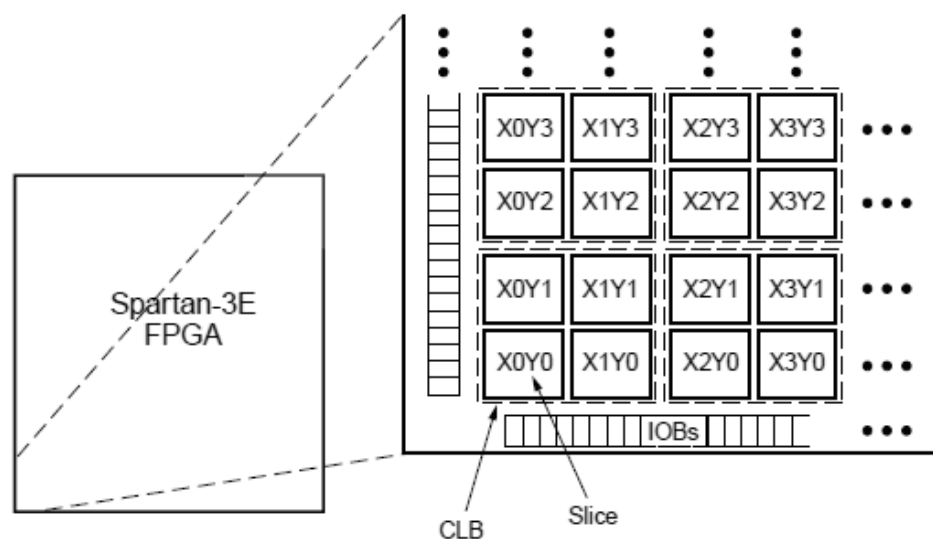
Pagal 11 paveikslėlį matyti, kad kiekvienas CLB blokas yra sudarytas iš keturių loginių dalelių (angl. „Slices“), kurios organizuotos stulpeliais ir sugrupuotos poromis. Paveikslėlyje esanti konfigūruojamo loginio bloko CLB kairiojo stulpelio loginių dalelių pora XILINX FPGA technologijoje vadinama SLICEM, o dešinioji stulpelio pora SLICEL. Šių dviejų porų skirtumas yra tas, kad SLICEM loginių dalelių poros LUT elementai gali būti naudojami kaip RAM16 arba SRL16 (SLICEM tipo loginių dalelių pora pavaizduota 11 pav. dešinėje pusėje), o SLICEL tokios galimybės neturi. Režimuojant CLB loginių dalelių porų (SLICEL ir SLICM) sandarą, apibrėžiame tikslų elementų kiekį bei tipą:

- Du keturių įėjimų LUT (Logic Up Table) loginių funkcijų lentelė;
- Du saugojimo elementai (registrai);
- Du plataus funkcionalumo skirstymo elementai (MUX);
- Loginiai elementai aritmetinėms funkcijoms atlikti (kiekis priklauso nuo algoritmo).

SLICEL poros LUT funkcijų generatorius gali būti programuojamas į:

- Du 16x1 paskirstytus RAM blokus, RAM16;
- Du 16 bitų poslinkio registrus, SRL16.

XILINX kompanijos FPGA matricose CLB bloką loginės dalelės (angl. „Slices“) yra numeruojamos eilučių ir stulpelių numeracijos pagalba. CLB loginių dalelių vietai nustatyti naudojama „X“ ir „Y“ ašys. „X“ ašies pagalba nustatoma, kuriame stulpelio vietoje randasi ieškoma loginė dalelė, o „Y“ ašis identifikuoja, kurioje eilutėje yra ieškoma CLB konfigūruojamų loginių bloką dalelė. Ašių numeracija prasideda nuo FPGA matricos kairiojo apatinio kampo (numeruojant nuo 0...1...2...ir t.t.). Žemiau esančiame paveikslėlyje matyti XILINX Spartan šeimos FPGA matricos fragmentą su loginių dalelių numeracija.



12 pav. FPGA matricos CLB bloką loginių dalelių numeracija

Šaltinis: „Spartan-3 Generation FPGA User Guide“ techninė dokumentacija parengta XILINX

Žemiau esančioje lentelėje pateikta XILINX Spartan šeimos FPGA matricų duomenys (ištekliai). Šioje lentelėje atsispindi FPGA matricos architektūros visi sudedamieji komponentai bei kiekiai (tam tikrai įrangai):

ĮRANGA	CLB eilė	CLB stulpelis	CLB viso	Loginės dalelės	LUT/ Flip-Flop	Loginės celės	RAM16/ SRL16	Paskirstyti RAM bitai
SPARTAN-3A DSP FPGA CLB ištekliai								
XC3SD1800A	88	48	4,160	16,640	33,280	37,440	16,640	266,240
XC3SD3400A	104	58	5,968	23,872	47,744	53,712	23,872	381,952
SPARTAN-3A/3AN FPGA CLB ištekliai								
XC3S50A/AN	16	12	176	704	1,408	1,584	704	11,264
XC3S200A/AN	32	16	448	1,792	3,584	4,032	1,792	28,672
XC3S400A/AN	40	24	896	3,584	7,168	8,064	3,584	57,344
XC3S700A/AN	48	32	1,472	5,888	11,776	13,248	5,888	94,208
XC3S1400A/AN	72	40	2,816	11,264	22,528	25,344	11,264	180,224
SPARTAN-3E FPGA CLB ištekliai								
XC3S100E	22	16	240	960	1,920	2,160	960	15,360
XC3S250E	34	26	612	2,448	4,896	5,508	2,448	39,168
XC3S500E	46	34	1,164	4,656	9,312	10,476	4,656	74,496
XC3S1200E	60	46	2,168	8,672	17,344	19,512	8,672	138,752
XC3S1600E	76	58	3,688	14,752	29,504	33,192	14,752	236,032
SPARTAN-3 FPGA CLB ištekliai								
XC3S50	16	12	192	768	1,536	1,728	768	12,288
XC3S200	24	20	480	1,920	3,840	4,320	1,920	30,720
XC3S400	32	28	896	3,584	7,168	8,064	3,584	57,344
XC3S1000	48	40	1,920	7,680	15,360	17,280	7,680	122,880
XC3S1500	64	52	3,328	13,312	26,624	29,952	13,312	212,992
XC3S2000	80	64	5,120	20,480	40,960	46,080	20,480	327,680
XC3S4000	96	72	6,912	27,648	55,296	62,208	27,648	442,368
XC3S5000	104	80	8,320	33,280	66,560	74,880	33,280	532,480

1 lent. XILINX SPARTAN šeimos FPGA matricų ištekliai

Šaltinis: „Spartan-3 Generation FPGA User Guide“ techninė dokumentacija parengta XILINX

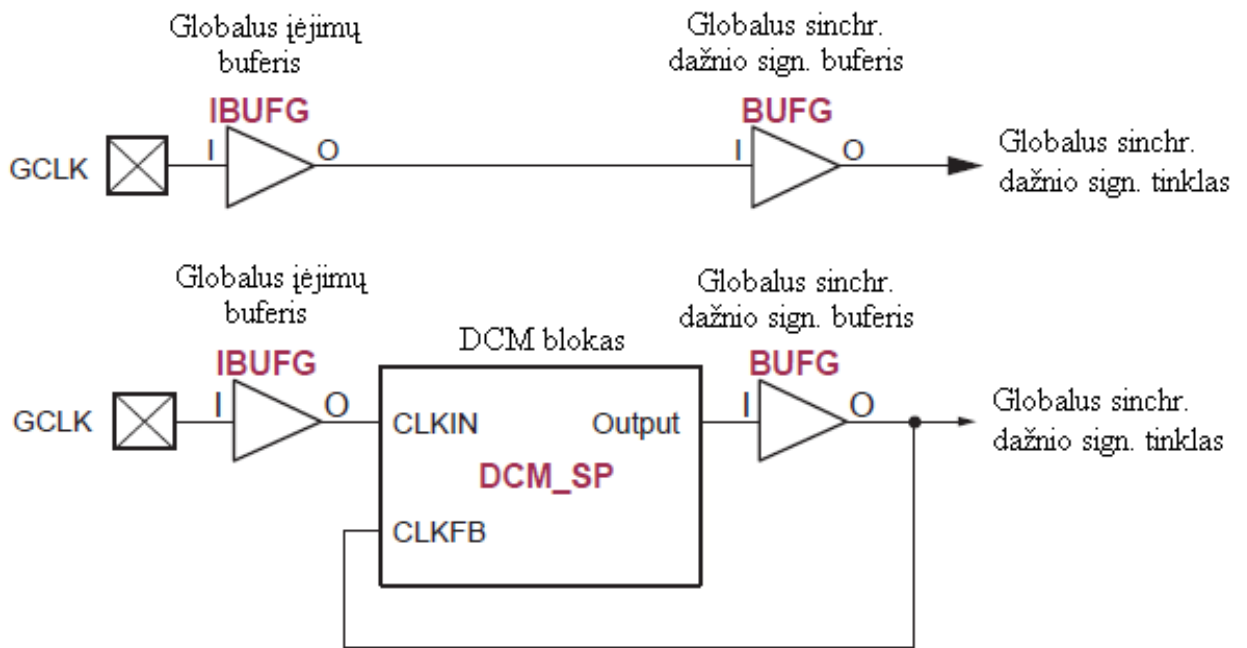
DCM (Digital Clock Management) – skaitmeninio sinchroninio dažnio signalo valdymo blokas. DCM paskirtis – paskirstyti generuojamą sinchroninio dažnio signalą, kiekvienam loginiam blokui. DCM pagalba pašalinami galimi sinchroninio signalo nukrypimai, taip pagerinant sistemos charakteristikas. Bei (priklusomai nuo projektuojamos situacijos) šis blokas padalina arba padaugina ateinantį sinchroninio dažnio signalą ir paskirsto vidiniams FPGA blokams (tokiems kaip CLB). Taigi DCM blokas su sinchroninio dažnio signalo tinklu integruotas yra tiesiogiai FPGA matricoje.

Kitais žodžiais tariant DCM integruoja suprogramuotą skaitmeninio sinchroninio dažnio signalą į FPGA vidinį GCDN (Global Clock Distribution Network) tinklą, t.y. į globalius sinchroninio dažnio kelius. Šis blokas padeda išspręsti aukšto dažnio bei didelių charakteristikų sistemose dažniausiai išskylančias problemas. Problemų sprendimo būdai:

- Pašalinami skaitmeninio sinchroninio dažnio signalo galimi nukrypimai. Tokiu būdu pagerinamos visapusiškos sistemos charakteristikos, padidinamas stabilumas ir išvengiama nereikalingas signalų vėlavimas;
- Nustatomas fiksuotas fazės poslinkio dydis sinchroninio dažnio signalo periodui;
- Padalijamas ar padauginamas įeinančio sinchroninio signalo dažnis arba sintezuojamas visiškai naujas dažnis, pagal įeinančio signalo dažnio dalybos ir daugybos operacijas;
- Užtikrinamas taisyklingas išeinančio sinchroninio dažnio signalo periodas;
- Atliekamas įeinančio sinchroninio dažnio signalo konvertavimas skirtingiems I/O (įėjties/išėjties) blokų standartams. Pavyzdžiui konvertuojamas ir nukreipiamas įeinantis LVTTTL sinchroninio dažnio signalas į LVDS;
- Kiekvienas iš išvardintų sprendimo būdų atliekamas tuo pačiu metu (lygiagrečiai).

Įvairiuose FPGA matricose DCM blokų vietos bei kiekis yra skirtingas. Pavyzdžiui XILINX Spartan šeimos FPGA matricose galima rasti nuo dviejų iki aštuonių DCM blokų. Galime teigi, kad kiekis priklauso nuo matricos dydžio (t.y. pagrinde nuo CLB blokų skaičiaus). Taigi XILINX kompanijos mažiausioms FPGA matricoms (pavyzdžiui Spartan – 3A DSP FPGA, tipas: XCS50A/AN ir pan.) naudojama architektūra su dviem DCM blokais. Vidutinių gabaritų FPGA turi 4 DCM blokus (tokios kaip Spartan – 3E FPGA, tipas: XC3S250E ir pan.), o didžiausios – 8 DCM blokus (pavyzdžiui, Spartan – 3 FPGA, tipas: XCS3S2000). Visi FPGA matricoje esantys skaitmeninio sinchroninio dažnio signalo valdymo blokai (DSC) yra sujungti vidiniu globaliniu sinchroninio dažnio signalo tinklu GCDN (Global Clock Distribution Network).

DCM blokai nėra privalomi elementai – tai pasirenkamieji (pagal vykdomo projekto reikalavimus). Jeigu projekte reikalingos aukščiau išvardintos DCM bloko galimybės, tada FPGA matricoje yra programuojamos DCM blokų funkcijos. Dažniausiai skaitmeninio sinchroninio dažnio signalo valdymo blokai (DSC) yra naudojami itin sudėtinguose sistemose, kur reikalingas labai aukštas sinchroninio dažnio tikslumas. Nenaudojant DCM blokų, įeinantis sinchroninio dažnio signalas tiesiogiai nukreipiamas į sinchroninio dažnio signalo globalinį tinklą GCDN (Global Clock Distribution Network) per globalų įėjimo buferį ir sinchroninio signalo buferį, kurie yra vidiniai GCDN tinklo elementai. Jeigu sistema naudoja DCM blokus, tada tarp šių buferių įterpiamas DCM blokas, kuris atlieka jam sukonfigūruotas (suprogramuotas) funkcijas (žiūr. 13 pav.). Taigi žemiau esančiame paveikslėlio viršutinėje dalyje matyti GCDN tinklo fragmentas be DCM bloko, o žemiau esančioje paveikslėlio dalyje GCDN su DCM bloku:



13 pav. FPGA matrica GCDN tinklo fragmentas (ne)naudojant DCM bloką

Šaltinis: „Spartan-3 Generation FPGA User Guide“ techninė dokumentacija parengta XILINX

Skaitmeninio sinchroninio signalo dažnio valdymo blokas susideda iš keturių elementų, kuriems priskirtos skirtingos funkcijos. Šie elementai yra:

- Vėlinimo skląsties kilpa DLL (Delay Locked Loop);
- Skaitmeninio signalo dažnio sintezatorius DFS (Digital Frequency Synthesizer);
- Fazės poslinkis PS (Phase Shift);
- Logikos indikatorius SL (Status Logic);

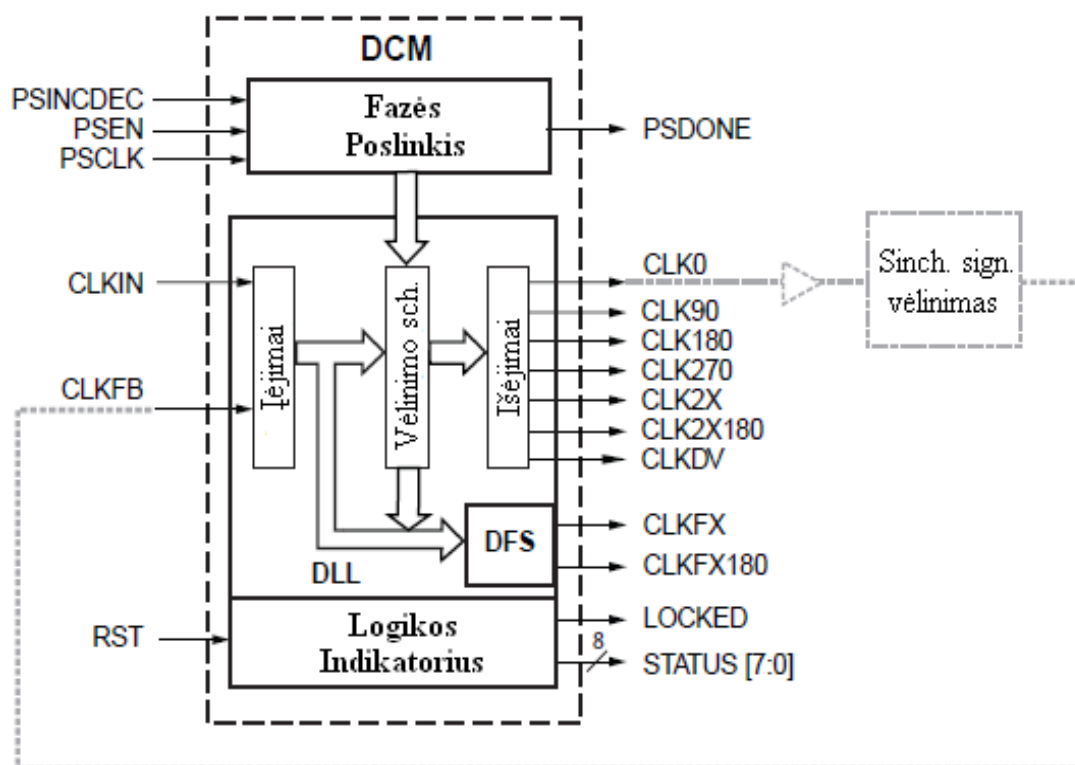
Taigi žemiau matysite kiekvieno elemento aprašą bei DCM funkcinę blokinę schemą (žiūr. 14 pav).

Vėlinimo skląsties kilpa DLL (Delay Locked Loop) – šio funkcinio elemento pagrindinė paskirtis – užtikrinti patikimą, be nukrypimų sinchroninio dažnio skaitmeninį išėjimo signalą. Elementas sudarytas iš skaitmeninių nukrypimų grandinės, kuri skirta efektyviai generuoti sinchroninio dažnio signalui (angl. „clock“) išvengiant vėlinimų bei signalo nukrypimų. Buferio pagalba DLL elementas sėkmingai eliminuoja sinchroninio dažnio įėjimo signalų vėlinimus individualiems FPGA blokams. Taigi, globalus sinchroninio signalo tinklo (GCDN) buferis priima sinchroninius signalus ir atitinkamai išvengdamas vėlinimų perduoda individualiems FPGA blokams. Žemiau esančiame paveikslėlyje (žiūr. 14 pav.) matyti du įeinančius sinchroninio dažnio signalus (CLKIN, CLKFB), kuriuos DCM bloko DLL elementas apdorojęs, perduoda identiškus signalus individualiems FPGA matricos blokams (globalaus sinchroninio dažnio signalų tinklo GCDN pagalba). Matyti, kad išeinantys signalai iš DLL funkcinio elemento yra CLK0, CLK90, CLK180, CLK270, CLK2X, CLK2X180, CLKDV.

Skaitmeninio signalo dažnio sintezatorius DFS (Digital Frequency Synthesizer) – skirtas nustatyti sinchroninio signalų išėjimo dažnį. Dažnis nustatomas pagal vartotojo konfigūruojamus (programuojamus) sandaugos bei dalybos loginius elementus. DFS funkcinis elementas generuoja du išėjimo signalus (CLKFX ir CLKFM180).

Fazės poslinkis PS (Phase Shift) – skirtas kontroliuoti DCM bloko sinchroninio dažnio išėjimo signalų fazes pagal įėjimo signalą CLKIN. Fazės poslinkio funkcinis elementas atlieka visų DCM bloko sinchroninio dažnio išėjimo signalų (viso 9 vnt.) fazės poslinkį, pagal sinchroninio dažnio įėjimo signalo periodo fiksuotą reikšmę. Fiksuota fazės poslinkio reikšmė yra nustatoma projektavimo metu, atliekant FPGA matricos konfigūravimo darbus. Fazės poslinkio funkcija gali būti atliekama skirtingais laiko intervalais (tai priklauso nuo FPGA matricos tipo bei konfigūravimo rezultatu). Pavyzdžiui XILINX kompanijos Spartan – 3 FPGA fazės poslinkio funkciją atliekama 256 kartus per CLKIN sinchroninio dažnio skaitmeninio signalo periodą.

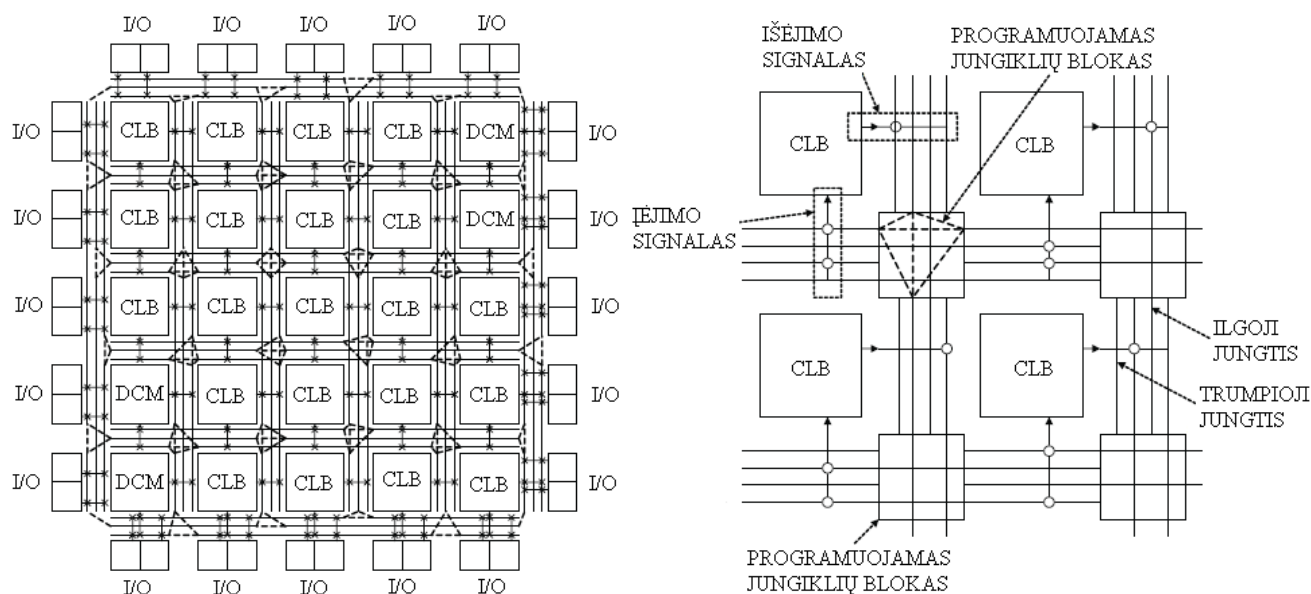
Logikos indikatorius SL (Status Logic) – šio funkcinio elemento paskirtis yra DCM bloko momentinių būsenų indikavimas. Indikacija atliekama naudojant išėjimo signalus (LOCKED, STATUS[0], STATUS[1] ir STATUS[2]). Išėjimo signalas LOCKED nusako DCM blokų išėjimo signalų fazės būsenas (lyginant su CLKIN įėjimo signalu). STATUS išėjimo signalai apibūdina DLL ir PS funkcinį elementų vykdomų operacijų būsenas.



14 pav. FPGA matricos DCM bloko funkcinė – blokinė schema

Šaltinis: „Spartan-3 Generation FPGA User Guide“ techninė dokumentacija parengta XILINX

Programuojamos vidinės jungtys (Interconnections). Šios vidinės jungtys tai programuojamas tinklas, kuris skirtas perduoti signalams iš vieno FPGA bloko išėjimų į sekančio bloko įėjimą. Atliekant sintezę, vidinės jungtys yra sujungiamos su FPGA matricos aptartais blokais: IOB (Input/Output block) – konfigūruojamas I/O (įėjimas/išėjimas) blokai, CLB (Configurable Logic Blocks) – konfigūruojamas loginis blokai, DCM (Digital Clock Management) – skaitmeninio sinchroninio dažnio signalo valdymo blokai bei SRAM blokai. Programuojamos vidinės jungtys sudarytos iš programuojamų jungiklių blokų (angl. „Switch Matrix“) bei laidžių jungčių (angl. „wires“), todėl FPGA matricos yra labai lanksčios projektų realizacijų atveju. Jungčių pagalba perduodamas signalas iš vieno FPGA vidinio bloko išėjimo į kito bloko įėjimą, o programuojamų jungiklių pagalba parenkamas signalo kelias, sujungiant horizontaliąsias ir vertikaliąsias jungtis tarpusavyje (prieklausomai kokį adresatą signalas turi pasiekti).



15 pav. FPGA matricos programuojamos vidinės jungtys

Šaltinis: „FPGA Architecture: Survey and Challenges“, autoriai: Ian Koun, Russell Tessier ir Jonathan Rose

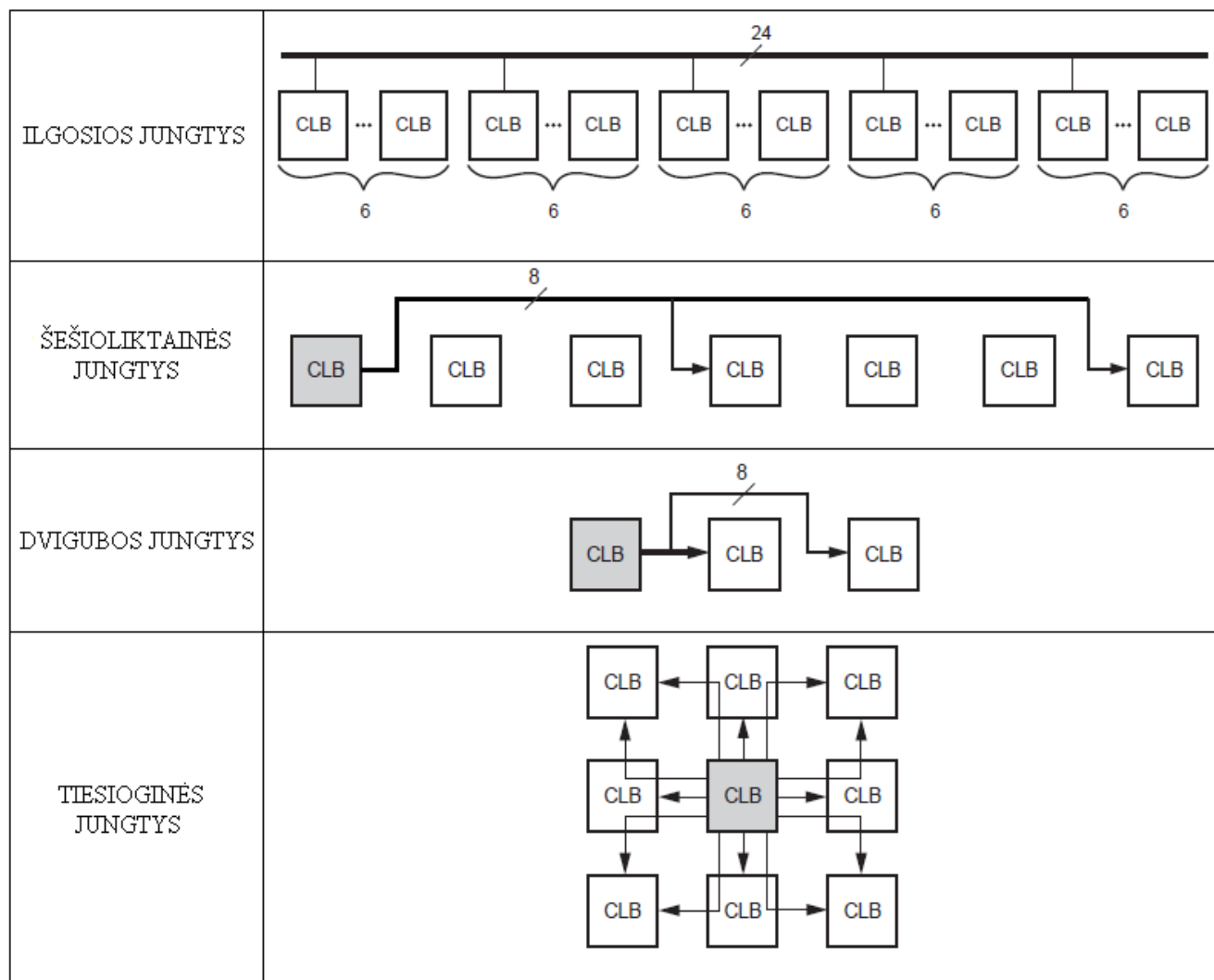
Valdymo sistemos realizuotos FPGA matricomis greitaveika labai priklauso nuo vidinių programuojamų jungčių ilgio. Jeigu FPGA matricoje esantys elementai yra išdėstyti neefektyviai (toli vienas nuo kito), atitinkamai programuojamos vidinės jungtys bus ilgos, to pasakoje signalo sklaidimo greitis – didelis. Tačiau, jeigu vidiniai FPGA matricos blokai išdėlioti pakankamai arti vienas kito, tada jungtys tarp šių blokų bus trumpos, to pasakoje signalo sklaidimo greitis – mažas.

XILINX kompanijos Spartan šeimos FPGA matricų programuojamos vidinės jungtys yra skirstomos į keturias rūšis:

- Ilgosios jungtys (angl. „Long Lines“);
- Šešioliktainės jungtys (angl. „Hex Lines“);

- Dvigubos jungtys (angl. „Double Lines“);
- Tiesioginės jungtys (angl. „Direct Lines“).

Žemiau esančioje lentelėje pateikti XILINX kompanijos Spartan šeimos jungčių tipai bei sujungimo būdai su vidiniais FPGA blokais:



2 lent. XILINX SPARTAN šeimos FPGA matricų programuojamų vidinių jungčių tipai

Šaltinis: „Spartan-3 Generation FPGA User Guide“ techninė dokumentacija parengta XILINX

1.5.2 MCU mikrovaldiklių RISC architektūros apžvalga

Autoriaus V. Bulovo knygoje „Mikroprocesoriai“ teigiama, kad mikrovaldiklių architektūra – tai pagrindinių jo įtaisų konfigūracija, jos ypatybės ir funkcinės galimybės duomenų apdorojimo, saugojimo ir persiuntimo aspektu. Taigi mikrovaldiklių architektūra apima kompleksą bendrų sistemos sudarymo klausimų, svarbių vartotojui, t.y. susijusių su sistemos galimybėmis, o ne techninės realizacijos detalėmis. Mikrovaldiklyje atliekamos visos valdymo ir informacijos apdorojimo operacijos, tačiau jis tik komponentas su kuriuo galima sudaryti kompleksinę valdymo sistemą.

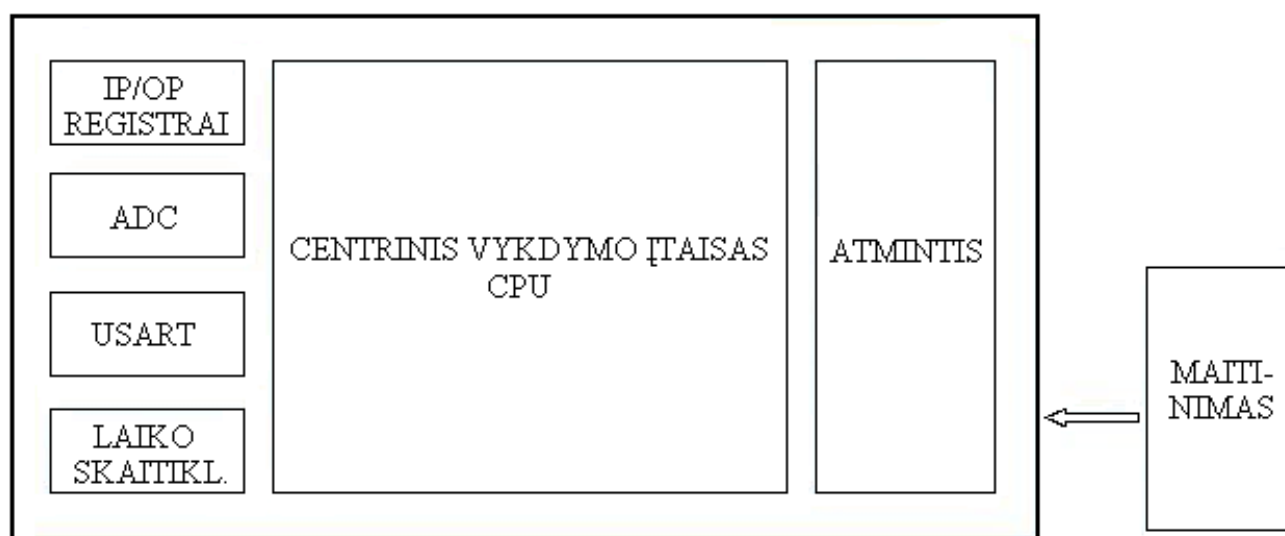
Mikrovaldiklis yra programuojamas įtaisas, taigi vartotojas turi galimybę sukurti sistemos valdymo algoritmą ir įrašyti į mikrovaldiklį (dažniausiai naudojamos ASM ir C programavimo kalbos).

Šiuo metu siūlomi įvairių kompanijų mikrovaldikliai, tarp jų populiariausi yra:

- Intel 8051 šeimos mikrovaldikliai;
- PIC mikrovaldikliai;
- AVR mikrovaldikliai.

Kadangi turėdami daugiau patirties su Atmel kompanijos gaminiais, todėl šiame darbe (projektinėje dalyje) tyrimui atlikti naudosime AVR šeimos mikrovaldiklių architektūrą (mikrovaldiklio tipas: Atmega103), todėl šiame skyrelyje plačiau panagrinėsime Atmel kompanijos AVR mikrovaldiklius.

Sekančioje knygoje „AVR RISC Microcontroller Handbook“, kurios autorius Claus Kuhnel teigiama, kad mikrovaldikliai sudaryti iš sudedamųjų elementų, tokių kaip: skaitymo – įrašymo atminties (Read – Write Memory) informacijos saugojimui, skaitymo atminties (Read – Only Memory), EEPROM – pastoviesiems duomenims saugoti, periferinių įtaisų bei I/O (įeičių/išeičių) sąsajos. Įvairių gamintojų mikrovaldiklių architektūros (pvz. RISC) elementai bei jų funkcionalumas yra vienodas arba labai panašus. Pavyzdžiui kiekvieno gamintojo gaminys turi atminties elementus, kurių paskirtis yra vienoda, t.y. komandų (instrukcijų) bei kitos informacijos saugojimas. Taip pat visuose gaminiuose naudojamo centrinio vykdymo įtaiso CPU (Central processing Unit) funkcija – programos srauto vykdymas/aparnavimas, informacijos valdymas/paskirstymas ir periferinių elementų, jungčių valdymas. Žemiau esančiame paveikslėlyje (žiūr. 16 pav.) pateikta bendra mikrovaldiklių blokinė struktūra:



16 pav. Bendroji mikrovaldiklių blokinė struktūra

Šaltinis: „AVR RISC Microcontroller Handbook“, autorius: Claus Kuhnel

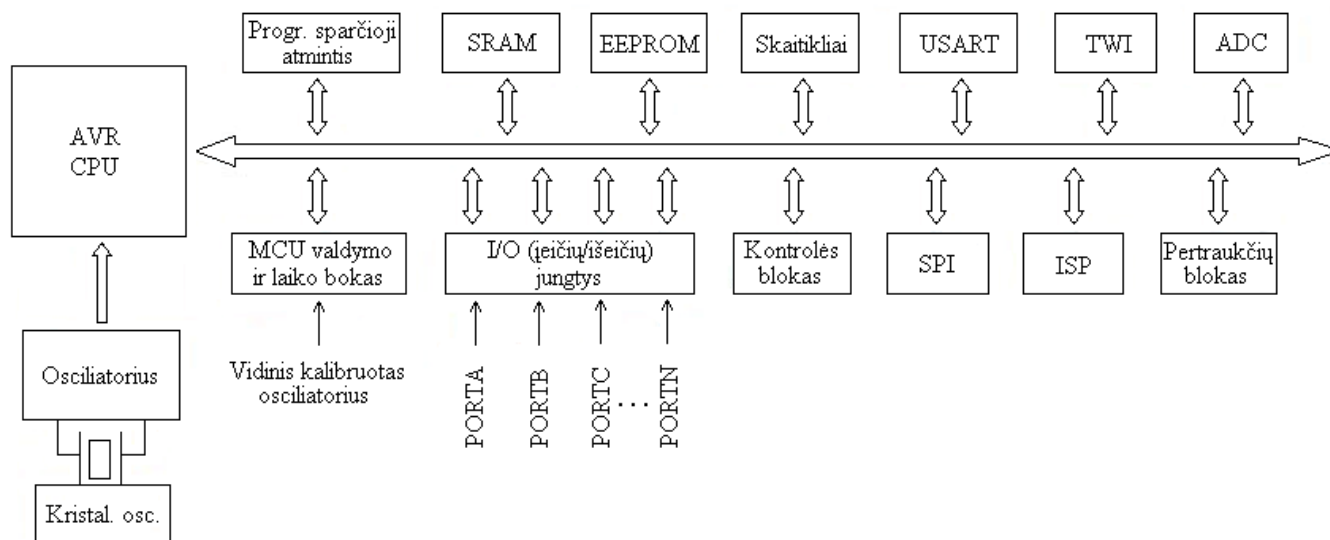
Paveikslėlyje matyti, kad centrinis vykdymo įtaisas CPU komunikuoja su visais elementais (funkciniais blokais). Kiekvienas mikrovaldiklių gamintojas norėdamas sukurti kuo našesnę mikrovaldiklį, stengiasi sumažinti bei optimizuoti CPU paskirtas užduotys, nepažeidžiant sistemos galimybių bei reikalavimų, tokiu būdu siekiama padidinti mikrovaldiklių veikimo greitaveiką. Tai yra įgyvendinama suteikiant šalutiniams funkciniais blokams tam tikras vykdymo užduotys. Taigi 16 paveikslėlio kairėje pusėje matyti suskirstyti periferiniai (šalutiniai) mikrovaldiklio elementai – funkciniai blokai. Šie funkciniai blokai sujungiami su išorinėmis sistemomis (procesais), todėl jų pagalba centrinis vykdymo įtaisas CPU yra mažiau apkrautas, nes periferiniai elementai perima dalį vykdymo užduočių, skirtų informacijai perduoti/priimti, procesams stebėti bei valdyti. Pavyzdžiui, naginėjant komunikaciją tarp mikrovaldiklio ir išorinių prietaisų, galima teigti, kad mikrovaldiklio centrinio vykdymo įtaiso CPU užduotis yra tik perduoti informaciją į nuoseklaus ryšio funkcinį elementą, kuris be CPU pagalbos atliks informacijos padalijimo (paskirstymo, rūšiavimo) bei sinchronizavimo funkcijas. Analogiškai priimant informaciją CPU atlieka paskirtas užduotys tik tada, kai informacijos bitai yra patalpinti buferyje. Taigi priimant informaciją CPU įtaisiui lieka priimti informacinius bitus iš vidinio mikrovaldiklio buferio ir atlikti pavestas funkcijas pagal suprogramuotą valdymo algoritmą. Informacijos perdavimui ir priėmimui asinchroninio ir sinchroninio nuoseklaus ryšio funkciniai elementai naudoja funkcinis blokus. Asinchroniniam ir sinchroniniam nuoseklaus ryšiui užtikrinti naudojamas UART (Universal Synchronous Asynchronous Receiver Transmitter) blokas sudarytas iš registrų. Sekantis funkcinis blokas (pavaizduotas 16 paveikslėlyje) yra ADC keitiklis. Šio keitiklio pagalba analoginis signalas yra konvertuojamas į diskretinį signalą. Konvertavus išorinį signalą į diskretinį, vykdomas informacijos perdavimas į vidinį mikrovaldiklio buferį, iš kurio CPU priima šį konvertuotą signalą.

Bendrosios mikrovaldiklių blokinės struktūros dešinėje pusėje (žiūr. 16 pav.) matyti centrinį vykdymo įtaisą CPU bei atminties bloką. Atminties blokas susideda iš skirtingu atminties funkcinį elementų, tokiu kaip programinė atmintis arba informacijos saugojimo atmintis. CPU ir atminties blokas vienas su kitu sujungti magistrale.

Atmel kompanijos AVR 8 – bitų mikrovaldikliai pagaminti remiantis AVR RISC (Reduced Instruction Set Computer) procesorių architektūra, kuri pasižymi paprastesne komandų seka lyginant su CISC (Complex Instruction Set Computer). Kadangi naudojant RISC architektūrą nereikia kurti sudėtingo komandų rinkinio, galima tiksliai prognozuoti komandų vykdymo laiką (mikrovaldiklio CPU ciklais), tokiu būdu inžinieriams lieka daugiau galimybių optimizuoti mikrovaldiklius, ypač – nuoseklų komandų vykdymą. RISC komandos paprastai būna vieno ilgio, kiekviena komanda turi tik vieną arba du operandus. RISC mikrovaldikliuose naudojama daug bendrosios paskirties registrų, sudarančių registrų langus. Adresų galimybės tokiuose mikrovaldikliuose paprastai būna smarkiai

apribojamos: esant reikalui, tam tikras atminties puslapis perkeliamas į registrų langą arba atvirkščiai, tačiau komandų, tiesiogiai dirbančių su atmintyje esančiais duomenimis, nėra.

Žemiau esančiame paveikslėlyje (žiūr. 17 pav.) matyti Atmel kompanijos AVR Atmega šeimos mikrovaldiklio architektūros blokinę schemą:



17 pav. Atmel AVR Atmega šeimos mikrovaldiklio architektūros blokinė schema

Šaltinis: „AVR RISC Microcontroller Handbook“, autorius: Claus Kuhnel

Paveikslėlyje (žiūr. 17 pav.) matyti šiuos Atmega šeimos mikrovaldiklių architektūros elementus:

- **Vidinis kalibruotas osciliatorius.** Atmega šeimos mikrovaldikliai turi kalibruotą vidinį sinchroninio signalo generatorių (osciliatorių), kuris prijungtas prie MCU valdymo ir laiko bloko. Įprastai šis signalas yra generuojamas 1MHz dažniu, tačiau turi galimybę būti nustatytas iki 8 MHz. Taip pat Atmega mikrovaldikliai gali būti prijungti prie išorinio osciliatoriaus, tokiu būdu sinchroninio signalo dažnį būtų galima pakelti iki 16MHz (priklausomai nuo Atmega mikrovaldiklio tipo);
- **ADC sąsaja.** Dauguma Atmega mikrovaldiklių turi 8 kanalų ADC (angl. „Analog to Digital Converter“) sąsają, kurio pagalba analoginis signalas konvertuojamas į 8 bitų skaitmeninį signalą;
- **Skaitikliai.** Atmega šeimos mikrovaldikliai turi 8 ir 16 bitų laiko skaitiklius, kurie naudojami generuoti tikslioms (laiko atžvilgiu) funkcijoms, pavyzdžiui laiko vėlinimui tarp dviejų operacijų;
- **Kontrolės blokas.** Šis blokas skirtas stebėti CPU valdymo algoritmų vykdymą bei esant sutrikimams (pvz.: valdymo komandos strigimas) perkrauti CPU įtaisą;
- **Pertraukčių blokas.** Atmega šeimos mikrovaldikliai turi tam tikrą kiekį (priklausomai nuo mikrovaldiklio tipo) pertraukčių, kurių pagalba sustabdomas programos vykdymas ir

pereinamas prie kitos operacijos. Atlikus šią operaciją tęsiamas pertrauktas programos vykdymas;

- **USART (Universal Synchronous and Asynchronous Receiver and Transmitter).** Sąsaja, kurios pagalba atliekama komunikacija su išoriniais įrenginiais nuosekliai perduodant/priimant bitus (angl. „bit by bit“);
- **I/O (įėjimų/išėjimų) jungtys.** Tai programuojamos skaitmeninių įėjimų/išėjimų jungtys, kurių pagalba priimami arba perduodami skaitmeniniai signalai (mikrovaldiklio maitinimo įtampos Vcc lygio) į/iš mikrovaldiklio į/iš išorinius prietaisus.
- **Programuojama sparcioji atmintis.** Šioje atmintyje yra saugomas programuojamas valdymo algoritmas (programa). Šią atmintį galima perrašyti pagal vartotojo poreikius, programuojant. Be to ši atmintis yra pastovi, t.y. dingus mikrovaldiklio maitinimui – duomenys yra išlaikomi;
- **EEPROM.** Tai taip pat pastovioji atmintis (t.y. dingus maitinimui, duomenys išlaikomi), tačiau jinais skirta saugoti įvairių kintamųjų (vykdant valdymo algoritmą) reikšmes;
- **SRAM (Static Random Access Memory).** Tai nepastovi mikrovaldiklio atmintis (t.y. dingus maitinimui atmintis ištrinama). Šią atmintį naudoja centrinis vykdymo įtaisas (CPU) valdymo algoritmo vykdymo momentu;
- **ISP (In System Programming).** ISP pagalba galima atlikti mikrovaldiklio programavimą neišimant jo iš valdymo sistemos grandinės;
- **SPI (Serial Peripheral Interface).** Sąsaja naudojama realizuoti komunikacijai tarp dviejų įtaisų, kurie naudoja vienodą generuojamą sinchroninio dažnio signalą;
- **TWI (Two Wire Interface).** TWI naudojamas realizuoti tinklui tarp unikalius adresus turinčių įtaisų. Taip suteikiama galimybė lygiagrečiai perduoti ar priimti informaciją tam tikriems tinkle esantiems įtaisams.

Paminėta, kad Atmel kompanijos AVR mikrovaldikliai naudoja RISC architektūrą. Šiuo atveju (Atmega mikrovaldiklių) naudojami 32 registrai – vykdomi komandomis (instrukcijomis). Visi registrai yra tiesiogiai sujungti su ALU (angl. „Arithmetic Logic Unit“, ALU vykdymo procedūros yra suskirstytos į: aritmetikos, logikos ir veiksmų su bitais procedūras) įtaisu, kuris geba priimti du nepriklausomus registrus (vienos instrukcijos) sinchroninio signalo ciklo metu (vienu periodu). Atmel kompanijos AVR mikrovaldikliai naudoja skirtingas atmintis bei magistrales programų vykdymui ir informacijos apdorojimui (t.y. naudojamas „Hardvard“ modelis). CPU vykdant valdymo algoritmą, komandos (instrukcijos) yra paduodamos iš programinės mikrovaldiklio atminties į CPU. Taigi, kai instrukcija yra pabaigta vykdyti, tada programos atmintis perduoda sekančią instrukciją CPU įtaisui. Tokiu būdu valdymo algoritmo instrukcijos yra vykdomos su kiekvienu sinchroninio dažnio signalo ciklu (angl. „clock cycle“). Mikrovaldiklio programinė atmintis yra realizuota programuojamoje

sparčiojoje atmintyje. Šioje atmintyje Atlemt AVR komandos (instrukcijos) yra 16 bitų dydžio (t.y. talpinami du 8 bitų registrai), tai reiškia, kad programinės atminties adresas yra suteikiamas kiekvienai 16 bitų komandai.

2. VĖJO JĖGAINIŲ VALDYMO SISTEMOS REALIZACIJOS IR TYRIMO DALIS

2.1 Įvadinė dalis

Aukščiau esančioje valdymo sistemų analitinėje dalyje išnagrinėtos vėjo jėgainių valdymo sistemos. Pateiktos ne tik Lietuvos, tačiau ir pasauliniu mastu išskylančios problemos bei sprendimo būdai, įvertinta esamų vėjo elektrinių valdymo sistemų situacija rinkoje bei išnagrinėti jų privalumai ir trūkumai. Atsižvelgiant į aptartų temų iškeltas problemas, šioje dalyje atliktas tyrimas naudojant dvi skirtingas valdymo sistemas vėjo elektrinių kontrolei. Taigi darbo tyrimas susideda iš sistemų:

- Valdymo sistemos realizuotos FPGA matricoje;
- Valdymo sistemos realizuotos MCU mikrovaldiklio architektūros pagrindu – FPGA matricoje;

Abiejų sistemų realizavimo etape naudojama FPGA matrica, tačiau skirtumas tarp sistemų yra tai, kad pirmuoju atveju vėjo jėgainių valdymo algoritmas realizuotas tiesiogiai FPGA matricoje. Antruoju atveju FPGA matricoje įgyvendinta standartinio mikrovaldiklio architektūra (su sudedamaisiais funkciniais elementais), kurioje atliekamas vėjo jėgainių valdymo algoritmas. Taigi antruoju atveju gauname standartinę MCU pagrindu realizuotą valdymo sistemą, kuri pritaikoma vėjo elektrinių kontrolei vykdyti. Abiem sistemoms sukurtas vienodas vėjo jėgainių valdymo algoritmas, kuris realizuotas naudojant 4MHz sinchroninio dažnio signalą. Šio darbo tikslas išsiaiškinti ir praktiškai pateikti skirtumus (pranašukus bei trūkumus) tarp atliktų sistemos realizacijų, palyginti sistemų greitaveikos ir patikimumo parametrus bei apibrėžti optimalią valdymo sistemos architektūrą skirtingiems vėjo jėgainių tipams (t.y. komercinės paskirties ir namų ūkio paskirties vėjo elektrinėms).

Vėjo jėgainių valdymo sistemos realizacijos ir tyrimo dalies aprašymas suskaidytas skyriais, kur pirmame skyriuje „Įvadinė dalis“ (t.y ši skaitoma dalis) apibrėžti tiriamojo darbo metodai ir pristatyta vėjo jėgainių valdymo sistemos realizacijos ir tyrimo dalies struktūra. Antrajame skyriuje „Tiriamojo daro vykdymo eigos aprašymas ir metodai“ aprašyta tiriamojo darbo vykdymo eiga, pateikti realizuotų valdymo sistemų tipai bei techninės charakteristikos, aprašytas sistemose įgyvendintas valdymo algoritmas vėjo jėgainėms, pateiktos valdymo algoritmo būsenų diagramos bei sudaryta bendra valdymų sistemų struktūra. Trečiajame šios dalies skyriuje „Tiriamojo darbo realizacijų analizavimas“ pateikti bei aprašyti vėjo jėgainių valdymo sistemų (realizuotų FPGA matricoje ir MCU mikrovaldiklio architektūroje) gauti modeliavimo ir sintezės rezultatai, atliktas sistemų tarpusavio palyginimas pagal gautus duomenys.

2.2 Tiriamojo darbo vykdymo eigos aprašymas ir metodai

2.2.1 Tiriamojo darbo vykdymo etapai

Nagrinėjant bei analizuojant tiriamojo darbo aprašymą, pirmiausia reikia apibūdinti darbo eigą bei akcentuoti kriterijus, kuriais remiantis pasirinkta ši darbo kryptis. Taigi šio tiriamojo darbo vykdymo eiga susideda iš dviejų pagrindinių etapų:

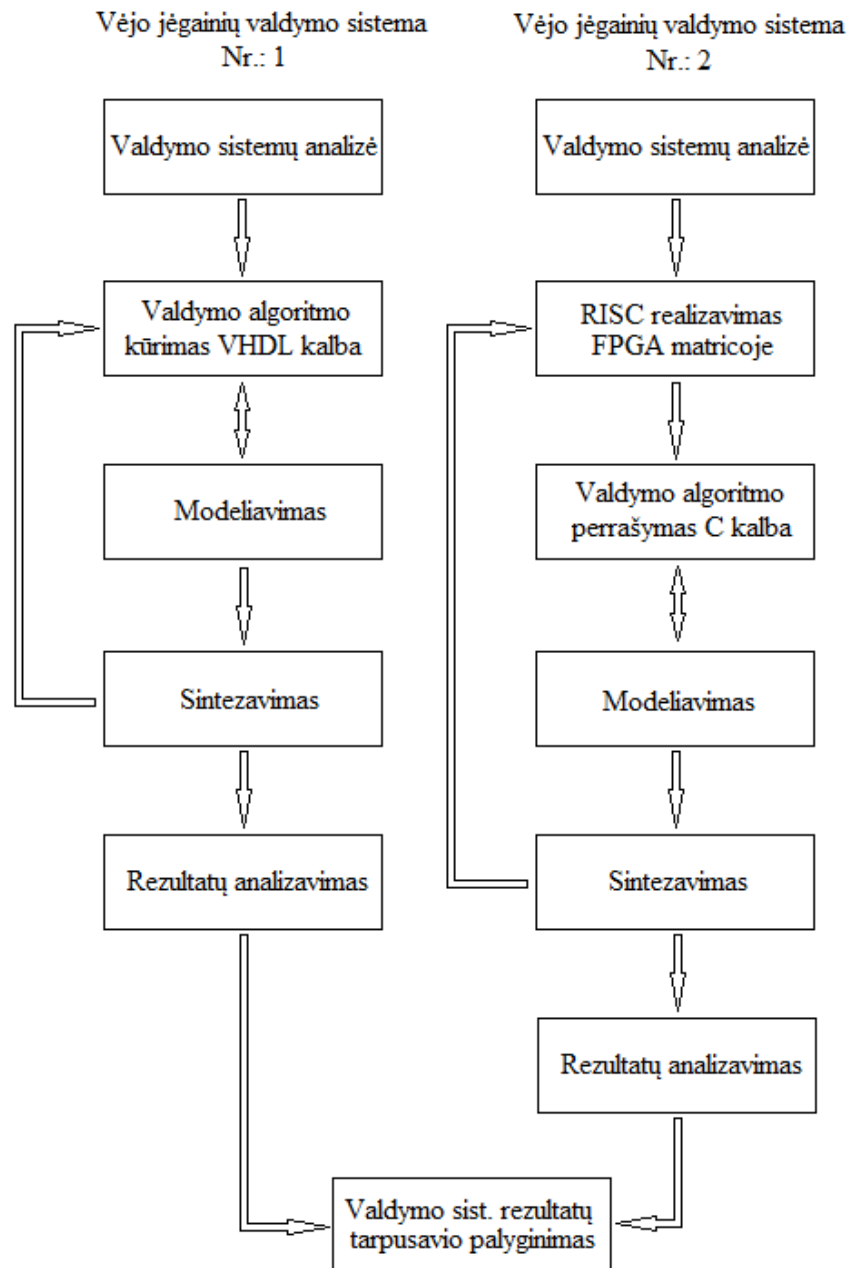
- vėjo jėgainių valdymo sistemos realizavimas;
- realizuotų vėjo jėgainių valdymo sistemų tyrimas.

Atsižvelgiant į aptartas valdymo sistemų analitinės dalies keliamas problemas, pirmajame tiriamojo darbo vykdymo etape pasirenkamos dvi skirtingos valdymo sistemų architektūros – tai FPGA matricos ir MCU (angl. Micro-controller Unit“) RISC architektūros. Šios dvi architektūros pasirinktos remiantis keliais kriterijais, t.y. tiriamajam darbu atlikti naudosime RISC architektūrą, nes tai viena iš populiariausių mikrovaldiklių architektūrų realizuotų valdymo sistemose (tame tarpe ir vėjo jėgainių valdymo sistemose). FPGA matricos pasirinktos, nes tai viena iš perspektyviausių, plačių galimybių, didelio lankstumo bei gana dideliais tempais populiarėjančių schemų, kurios realizuojamos įvairiuose sistemose. Detalesnė tiriamojo darbo eigos struktūra pateikta žemiau esančioje blokine schemoje (žiūr. 18 pav.). Šiame paveikslėlyje matyti dviejų vėjo jėgainių valdymo sistemų realizavimo ir tyrimo darbo eigos etapai. Kairėje paveikslėlio pusėje esančios sistemos Nr.:1 – FPGA matricos pagrindu realizuota vėjo elektrinių valdymo sistemos realizacijos ir tyrimo eiga. Sekanti vėjo jėgainių valdymo sistema Nr.: 2 (dešinėje paveikslėlio pusėje) – mikrovaldiklio RISC architektūros valdymo sistemos (įgyvendintos FPGA matricoje) realizacijos ir tyrimo eiga. Pirmosios sistemos (t.y. vėjo jėgainių valdymo sistema Nr.: 1) realizacijos ir darbų vykdymo etapų metu atliekamos šios dalys:

- Valdymo sistemų analizė;
- Valdymo algoritmo kūrimas VHDL kalba;
- Modeliavimas;
- Sintezavimas;
- Rezultatų realizavimas;
- Valdymo sistemų gautų rezultatų tarpusavio palyginimas.

Valdymo sistemos analizės dalyje tiriamos valdymo sistemos, jų techninės charakteristikos bei vėjo jėgainių valdymo sistemų valdymo algoritmai. Taigi šiame etape yra nagrinėjamos esamos vėjo jėgainių valdymo sistemos, kurios realizuotos FPGA matricos pagrindu, įvertinamos techninės charakteristikos bei išsirenkama FPGA matricos platforma, kurioje bus įgyvendintas sukurtas valdymo algoritmas.

Valdymo algoritmo kūrimo VHDL kalba dalyje išsirenkami dažniausiai pasitaikantys vėjo jėgainių valdymo sistemų valdymo būdai. Šie būdai aprašomi VHDL kalba bei atliekamas išeities kodo kompiliavimas. Tokiu būdu sudaromas valdymo algoritmas, skirtas vėjo elektrinių darbo kontrolei ir įvykių / momentinių reikšmių stebėjimui.



18 pav. Vėjo jėgainių valdymo sistemų realizacijos ir tyrimo dalies vykdymo eiga

Šaltinis: sudaryta autoriaus

Modeliavimo dalyje sudaromas testo vykdomo kodas (VHDL kalba) ir atliekamas vėjo jėgainės valdymo sistemos Nr.: 1 testavimas. Testo vykdomo kode aprašomos galimos vėjo jėgainės darbo būsenos ir stebima sistemos reakcija į šias reikšmes. Esant neatitikimams, grįžtama į prieš tai esantį

etapą, kuriame atliekamas valdymo algoritmo VHDL išeities kodo koregavimas. Ištestavus sistemą pereinama prie sekančios darbo dalies – sintezės.

Sintezavimo dalyje valdymo algoritmas yra sintezuojamas, tokiu būdu yra paruošiamas algoritmas realizavimui FPGA matricoje. Sintezės rezultato gavimo metu išeities kodas pateikiamas grafiniu atvaizdu, kuriame matyti sintezuoti loginiai elementai (sudaryti iš loginių ventilių) bei jungtys tarp jų. Naudojamų loginių elementų kiekis bei ryšiai tarp elementų priklauso nuo valdymo algoritmo funkcijų. Jeigu išeities kodas nėra sintezuojamas, tada grįžtama į dalį, kurioje atliekamas valdymo algoritmo kūrimas (šiuo atveju koregavimas) VHDL kalba, priešingu atveju (esant sintezuojamam išeities kodui) pereinama į sekančią rezultatų analizavimo dalį.

Rezultatų analizavimo dalyje naudojama programinė įranga (Xilinx ISE Project Navigator), kurios pagalba atliktos ne tik prieš tai buvusios dalys (valdymo algoritmo kūrimas, modeliavimas, sintezavimas), tačiau ir ši dalis – rezultatų analizavimas. Programinės įrangos pagalba stebima sistemos greitimeika, signalų sklidimo greitis loginiuose elementuose bei jungtyse tarp elementų, naudojamų loginių elementų tipas bei kiekis, vartojamos el. galios dydis ir pan.

Atlikus rezultatų analizavimą galima pereiti prie paskutinės tiriamojo darbo dalies – **valdymo sistemų rezultatų tarpusavio palyginimo**, kurioje lyginami abiejų realizuotų vėjo jėgainių valdymo sistemų gauti rezultatai bei formuluojamos išvados.

Antrosios sistemos (t.y. vėjo jėgainių valdymo sistema Nr.: 2) realizacijos ir darbų vykdymo etapų metu atliekamos šios dalys:

- Valdymo sistemų analizė;
- RSIC realizavimas FPGA matricoje;
- Valdymo algoritmų perrašymas C kalba;
- Modeliavimas;
- Sintezavimas;
- Rezultatų realizavimas;
- Valdymo sistemų gautų rezultatų tarpusavio palyginimas.

Valdymo sistemos analizės dalyje tiriamos valdymo sistemos, jų techninės charakteristikos bei vėjo jėgainių valdymo sistemų valdymo algoritmai. Šiame etape yra nagrinėjamos esamos vėjo jėgainių valdymo sistemos, kurios realizuotos mikrovaldiklių pagrindu (RISC architektūra), įvertinamos techninės charakteristikos bei išsirenkamas mikrovaldiklis, kurio galimybėmis remiantis sekančiuose dalyse bus įgyvendintas valdymo algoritmas. Papildomai šioje dalyje išsirenkama FPGA platforma, kurioje bus realizuota pasirinkto mikrovaldiklio architektūra (siekiant tikslingai atlikti tyrimą parenkama FPGA platforma identiška vėjo jėgainių valdymo sistemos Nr.: 1 platformai);

RSIC realizavimo FPGA matricoje dalyje tobulinama ir testuojama Ruslan Lepetenok sukurta mikrovaldiklio architektūros realizacija FPGA matricoje. Taigi šioje dalyje modeliuojamas bei

sintezuojamas mikrovaldiklio branduolio (architektūros struktūrą su funkciniais elementais galima rasti „Mikrovaldiklių (MCU) architektūros apžvalga“ skyrelyje) išeities kodas. Kadangi realizuojamos vėjo jėginių valdymo sistemos funkcijos neapėpia visų mikrovaldiklių galimybių, todėl FPGA matricoje realizuojami tik būtinausi (naudojami) Ruslan Lepetenok sukūrti (VHDL kalba) RISC mikrovaldiklio architektūros funkciniai elementai.

Valdymo algoritmo perrašymo C kalba dalyje atliekamas vėjo jėginių valdymo sistemos Nr.:1 VHDL išeities kodo perrašymas į C kalbą. Naudojant programinę įrangą „AVR Studio 4.16“ atliekamas sistemos kompiliavimas bei funkcinis valdymo algoritmo patikrinimas. Tinkamai realizuotas valdymo algoritmas, perkeliamas (naudojant konverterį „Hex to VHDL“) į FPGA matricoje realizuotą mikrovaldiklio programuojamos sparčiosios atminties elementą ir pereinama prie sekančios tiriamojo darbo dalies – modeliavimo.

Modeliavimo dalyje sudaromas testo vykdymo kodas (VHDL kalba) ir atliekamas vėjo jėginės valdymo sistemos Nr.: 2 testavimas. Testo vykdymo kode aprašomos galimos vėjo jėginės darbo būsenos ir stebima sistemos reakcija į šias reikšmes. Esant neatitikimams, grįžtama į prieš tai esantį etapą, kuriame atliekamas valdymo algoritmo C kalbos išeities kodo koregavimas. Ištestavus sistemą pereinama prie sekančios darbo dalies – sintezės.

Sintezavimo dalyje valdymo algoritmas yra sintezuojamas, tokiu būdu yra paruošiamas algoritmas realizavimui FPGA matricoje. Sintezės rezultato gavimo metu išeities kodas pateikiamas grafiniu atvaizdu, kuriame matyti sintezuoti loginiai elementai (sudaryti iš loginių ventilių) bei jungtys tarp jų. Naudojamų loginių elementų kiekis bei ryšiai tarp elementų priklauso nuo mikrovaldiklio architektūros struktūros. Jeigu išeities kodas nėra sintezuojamas, tada grįžtama į dalį, kurioje atliekamas RISC realizavimas FPGA matricoje (šiuo atveju koregavimas) VHDL kalba, priešingu atveju (esant sintezuojamam išeities kodui) pereinama į sekančią, rezultatų analizavimo dalį.

Rezultatų analizavimo dalyje naudojama programinė įranga (Xilinx ISE Project Navigator), kurios pagalba atliktos ne tik prieš tai buvusios dalys (RISC tobulinimas FPGA matricoje, modeliavimas, sintezavimas), tačiau ir ši dalis – rezultatų analizavimas. Programinės įrangos pagalba stebima sistemos greitimeika, signalų sklidimo greitis loginiuose elementuose bei jungtyse tarp elementų, naudojamų loginių elementų tipas bei kiekis, vartojamos el. galios dydis ir pan.

Atlikus rezultatų analizavimą galima pereiti prie paskutinės tiriamojo darbo dalies – **valdymo sistemų rezultatų tarpusavio palyginimo**, kurioje lyginami abiejų realizuotų vėjo jėginių valdymo sistemų gauti rezultatai bei suformuojamos išvadas.

Taigi šiame skyriaus „Tiriamojo darbo vykdymo eigos aprašymas ir metodai“ poskyryje aptarti tiriamojo darbo vykdymo etapai, o sekančiuose poskyriuose pateiktos realizuotų vėjo jėginių valdymo sistemų sudedamųjų dalių charakteristikos ir tipai, realizacijos metodai bei valdymo algoritmų būsenų diagramos.

2.2.2 Tiriamojo darbo metodai

Poskyryje „Tiriamojo darbo vykdymo etapai“ buvo pristatytos dvi skirtingos, tačiau tas pačias funkcijas atliekančias vėjo jėgainių valdymo sistemos. Šio poskyrio kontekste pateiktos platformų (FPGA matricos ir MCU mikrovaldiklio) techninės charakteristikos bei detaliau išnagrinėti realizuojamų vėjo jėgainių valdymo sistemų algoritmai.

Toliau remiantis „Tiriamojo darbo vykdymo etapai“ poskyrio vėjo jėgainių valdymo sistemų klasifikacija, detaliau aptariamos realizuotos sistemos bei pateikiamos jų technines charakteristikos:

Vėjo jėgainių valdymo sistema Nr.: 1 – tai sistema, kurios valdymo algoritmas realizuotas tiesiogiai FPGA matricoje. Ši vėjo elektrinių valdymo sistema turi galimybę priimti anemometro (vėjo greičio matuoklio) generuojamus impulsinius signalus (vienos sekundės intervale), nustatyti vėjo kryptį, sekti generuojamas įtampos momentinius duomenys bei stebėti baterijų bloko įkrovos/iškrovos momentines reikšmes. Pagal šiuos momentinius duomenys sistema atlieka vėjo jėgainės kampo tarp horizontaliosios turbinos ašies ir oro srauto krypties kontrolę, tinkamą baterijos bloko įkrovos/iškrovos reguliavimą bei apsaugą nuo kritinių vėjo greičių (t.y. apriojamas jėgainės rotoriaus sukimosi greitis, pasiekus 25 m/s). Visi momentiniai duomenys (vėjo greitis, kryptis, baterijų įkrovos/iškrovos bei generuojamos įtampos duomenys) perduodami vartotojui į PC (personalinį kompiuterį), naudojant UART (Universal Asynchronous Receiver Transmitter) protokolą. Taigi šios vėjo jėgainės valdymo sistemos pagrindiniai techniniai duomenys:

- FPGA matrica: SPARTAN – 6; tipas: XC6SLX4;
 - Loginių celių sk. (Logic Cells): 3840;
 - Loginės dalelių sk. (Slices): 600;
 - Trigerių sk. (Flip – Flop tipo): 4800;
 - RAM blokų sk. (18 Kb): 12;
 - DCM elementų sk.: 4;
 - I/O (įeičių/išeičių) blokų sk.: 4;
 - I/O (įeičių/išeičių) sk.: 132.
- ADC keitiklių sk.: 3;
- RS232 sąsaja: TX – RX;
- Osciliatorius: 50 MHz;
- Programavimo sąsaja: USB;

Vėjo jėgainių valdymo sistema Nr.: 2 – tai sistema, kurios valdymo algoritmas realizuotas MCU mikrovaldiklio RISC architektūros pagrindu. Mikrovaldiklio centrinis vykdymo įtaisas CPU su visais būtiniausiais funkciniais elementais (žiūr. 17 pav. Skyriuje: „Mikrovaldiklių MCU architektūros apžvalga“) įgyvendintas FPGA matricoje. Siekiant tyrimui sudaryti vienodas sąlygas skirtingoms

sistemoms (Nr.: 1 ir Nr.: 2), valdymo algoritmas bei galimybės išlaikomos vienodos. Taigi ši vėjo elektrinių valdymo sistema (kaip ir anksčiau aptartoji) turi galimybę priimti anemometro (vėjo greičio matuoklio) generuojamus impulsinius signalus (vienos sekundės intervale), nustatyti vėjo kryptį, sekti generuojamas įtampos momentinius duomenys bei stebėti baterijų bloko įkrovos/iškrovos momentines reikšmes. Pagal šiuos momentinius duomenys sistema atlieka vėjo jėgainės kampo tarp horizontaliosios turbinos ašies ir oro srauto krypties kontrolę, tinkamą baterijos bloko įkrovos/iškrovos reguliavimą bei apsaugą nuo kritinių vėjo greičių (t.y. apribojamas jėgainės rotoriaus sukimosi greitis, pasiekus 25 m/s). Visi momentiniai duomenys (vėjo greitis, kryptis, baterijų įkrovos/iškrovos bei generuojamos įtampos duomenys) perduodami vartotojui į PC (personalinį kompiuterį), naudojant UART (Universal Asynchronous Receiver Transmitter) protokolą.

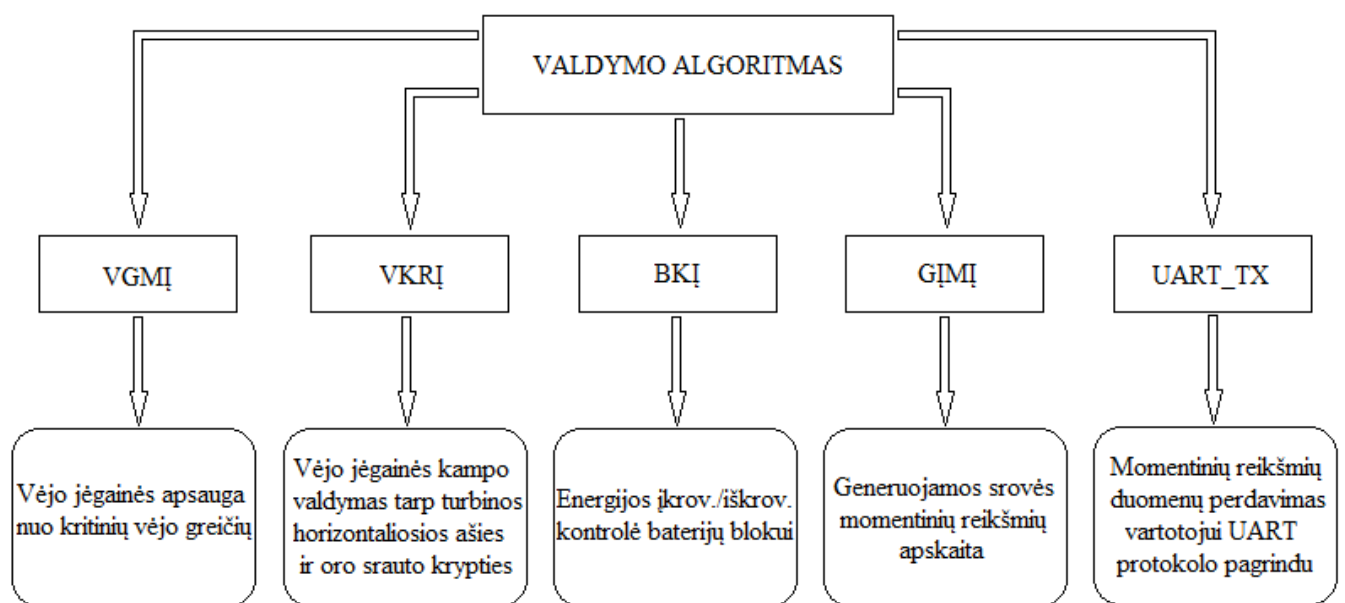
Ši realizacija susideda iš dviejų sistemų: FPGA matricos ir MCU mikrovaldiklio RISC architektūros, kuri įgyvendinta FPGA matricoje. Taigi pagrindiniai šios vėjo jėgainės valdymo sistemos techniniai duomenys:

- FPGA matrica: SPARTAN – 6; tipas: XC6SLX4;
 - Loginių celių sk. (Logic Cells): 3840;
 - Loginės dalelių sk. (Slices): 600;
 - Trigerių sk. (Flip – Flop tipo): 4800;
 - RAM blokų sk. (18 Kb): 12;
 - DCM elementų sk.: 4;
 - I/O (įeičių/išeičių) blokų sk.: 4;
 - I/O (įeičių/išeičių) sk.: 132 (programuojamos).
- ADC keitiklių sk.: 3;
- RS232 sąsaja: TX – RX;
- Osciliatorius: 50 MHz;
- Programavimo sąsaja: USB;
- MCU RISC architektūra: AVR Atmega103 mikrovaldiklio;
 - CPU įtaisas: 1;
 - Programinė atmintis: 128 KB;
 - SRAM atmintis: 4 KB;
 - I/O (įeičių/išeičių) sk.: 32 (programuojamos);
 - Laiko skaitiklių sk.: 1;
 - UART funkcija: 1;

Kaip paminėta anksčiau, abejoms vėjo elektrinių valdymo sistemoms yra realizuotas vienodas algoritmas, kuriems išeities kodas parašytas VHDL kalba (vėjo jėgainių valdymo sistemai Nr.: 1) ir C kalba (vėjo jėgainių valdymo sistemai Nr.: 2). Šio valdymo algoritmo funkcijas galime suskirstyti sekančiai:

- Vėjo jėgainės apsauga nuo kritinių vėjo greičių;
- Vėjo jėgainės kampo valdymas tarp horizontaliosios turbinos ašies ir oro srauto krypties;
- Energijos įkrovos/iškrovos kontrolė baterijų blokui;
- Generuojamos įtampos momentinių reikšmių apskaita (toliau GĮMĮ schema);
- Generuojamos įtampos, vėjo greičio, krypties bei baterijų bloko talpos momentinių duomenų išvedimas vartotojui naudojant UART protokolą.

Žemiau esančiame paveikslėlyje (žiūr. 19 pav.) matyti realizuoto algoritmo (valdymo sistemos) struktūrą:



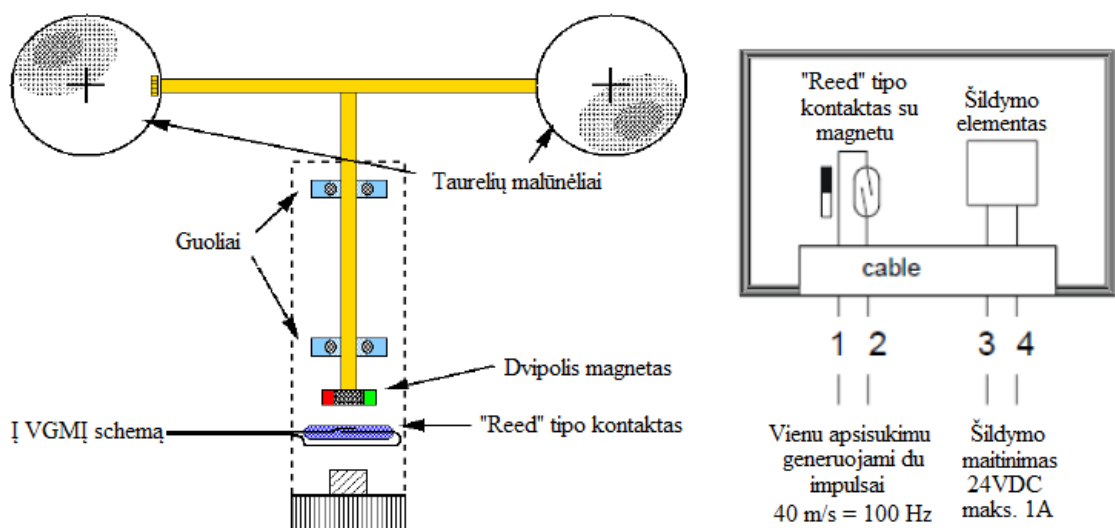
18 pav. Realizuotų vėjo jėgainių valdymo sistemų algoritmo struktūra

Šaltinis: sudaryta autoriaus

Viršuje esančiame paveikslėlyje pavaizduota, kad kiekvienai funkcijai atlikti yra sudaromos atskiros valdymo algoritmo dalys (schemos), pavyzdžiui pirmajai funkcijai (vėjo jėgainės apsaugai nuo kritinių vėjo greičių) atlikti yra atsakinga VGMĮ (Vėjo Greičio Matavimo Įtaiso) schema. Vėjo jėgainės kampo valdymo tarp horizontaliosios turbinos ašies ir oro srauto krypties funkciją įgyvendina VKRĮ (Vėjo Krypties Reguliavimo Įtaiso) schema. Trečiajai funkcijai (energijos įkrovos/iškrovos

kontrolė baterijų blokui) atlikti naudojama BKĮ (Baterijų Krovos Įtaiso) schema. Generuojamos srovės momentinių reikšmių apskaitos realizavimui sudaroma GĮMĮ (Generuojamos Įtampos Matavimo Įtaiso) schema bei visų kontroliuojamų vėjo jėgainės momentinių reikšmių duomenys perduodami vartotojui UART protokolo pagrindu, realizuojant UART_TX schemą.

VGMĮ (Vėjo Greičio Matavimo Įtaiso) schemas pagrindinė funkcija – matuoti vėjo greitį bei aptikus kritinį vėją (pvz. vėjui pasiekus 25 m/s) generuoti aukšto fronto signalą, kuris skirtas vėjo jėgainės rotorius apriboti. Vėjo matavimui naudojamas „reed“ tipo anemometras su taurelių malūnėliu. Pučiant vėjui šis prietaisas, suka taurelių malūnėlį, tokiu būdu magnetų ir „reed“ kontakto pagalba perduodami impulsiniai signalai FPGA matricoje projektuojamai VGMĮ schemai (vėjo jėgainės valdymo sistemos Nr.: 1 realizavimo atveju) arba Atmega103 mikrovaldiklio architektūros elementams, kurie įgyvendinti FPGA matricoje (vėjo jėgainės valdymo sistemos Nr.: 2 realizavimo atveju). Pagal impulso pasikartojimų skaičių per laiko vienetą (šiuo atveju sekundę), nustatomas impulsų dažnis. VGMĮ projektuojamos schemas vidinis skaičiavimo mechanizmas suskaičiuoja pasikartojančių impulsų skaičių per vieną sekundę. Tokiu būdu įvertinamas vėjo greitis, pagal kurį esant reikalui apribojamos rotorius apsisukimų momentas. Žemiau pateiktas anemometro bendras vaizdas (žiūr. 19 pav.) bei techninės charakteristikos:



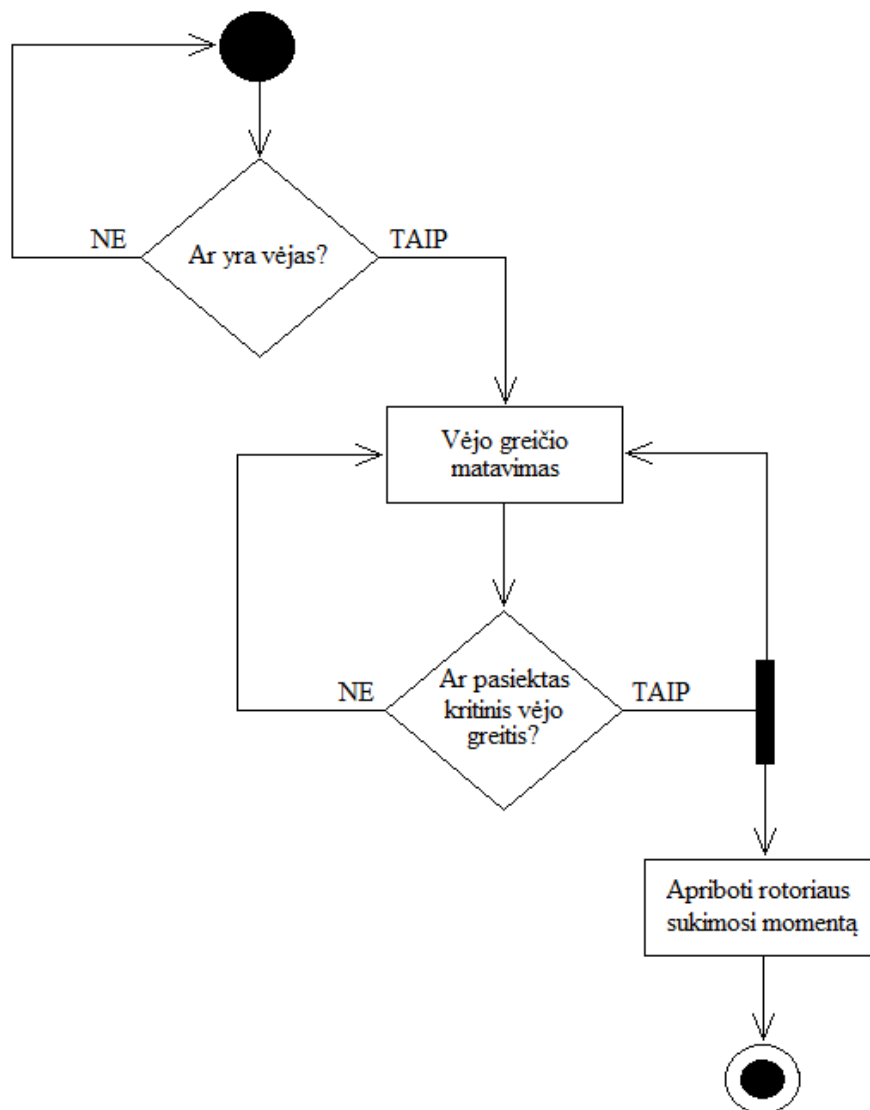
19 pav. Vėjo greičio matuoklio – anemometro bendras vaizdas

Šaltinis: sudaryta autoriaus

Anemometro pagrindinės techninės charakteristikos:

- Matavimo ribos: 0.5...40 m/s;
- Tikslumas: ± 0.5 m/s;
- Išėjimo signalas 0...100Hz;

Žemiau esančiame paveikslėlyje (žiūr. 20pav.) pateikta VGMĮ schemos būsenos diagrama, kurioje matyti šios schemos funkcionalumas. Taigi pirminėje funkcijos vykdymo būsenoje nustatoma ar yra vėjas, kai vėjas yra didesnis nei 0.5 m/s tada atliekamas vėjo greičio matavimas. Šiame funkcijos vykdymo etape matuojamas anemometro generuojamo impulsinio signalo skaičius vienos sekundės intervale, tokiu būdu įvertinamas momentinis vėjo greitis. Jeigu vėjo greitis didesnis už kritinę vertę (šiuo atveju 25 m/s) tada generuojamas aukšto fronto signalas, kurio pasekoje yra apribojamas rotoriaus sukimosi momentas (vėjo greičio matavimas nenutraukiamas). Priešingu atveju (kai vėjo greitis nesiekia kritinės vertės) veiksmų nesiimama, toliau vykdoma vėjo greičio matavimo funkcija.

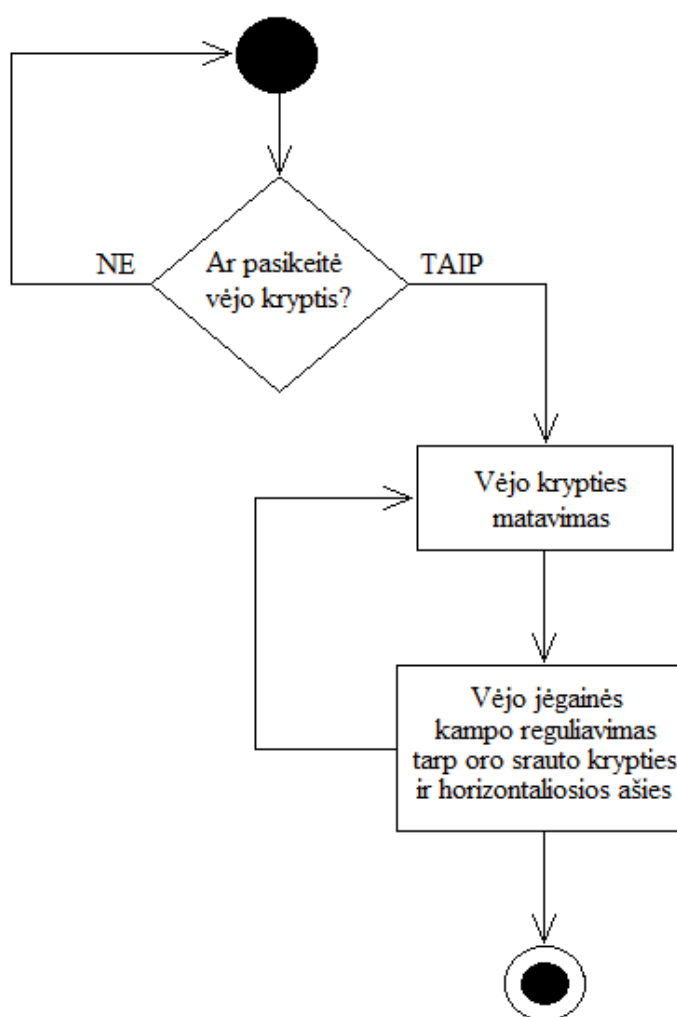


20 pav. VGMĮ schemos būsenos diagrama

Šaltinis: sudaryta autoriaus

VKRĮ (Vėjo Krypties Reguliavimo Įtaiso) schemos pagrindinės funkcijos – matuoti vėjo kryptį bei reguliuoti vėjo jėgainės kampą tarp oro srauto krypties ir horizontalios turbinos ašies.

Pirmajai funkcijai atlikti naudojamas vėjo krypties matavimo prietaisas, kuris pučiant vėjui sukasi pagal vėjo kryptį $0...360^\circ$ intervale, atitinkamai generuodamas 0-5V įtampą. Ši generuojama įtampa nukreipiama į ADC keitiklį iš kurio skaitmeniniai signalai perduodami FPGA matricoje projektuojamai VKRĮ schemai (vėjo jėgainės valdymo sistemos Nr.: 1 realizavimo atveju) arba Atmega103 mikrovaldiklio architektūros elementams, kurie įgyvendinti FPGA matricoje (vėjo jėgainės valdymo sistemos Nr.: 2 realizavimo atveju). VKRĮ schemeje realizuotas ADIA (Analog Digital Interface Adapter) elementas, kuris geba priimti ADC siunčiamus signalus. Sekančiai funkcijai atlikti naudojamas algoritmas pagal kurį atliekamas žingsninio variklio valdymas (kampo reguliavimui pagal vėjo kryptį).



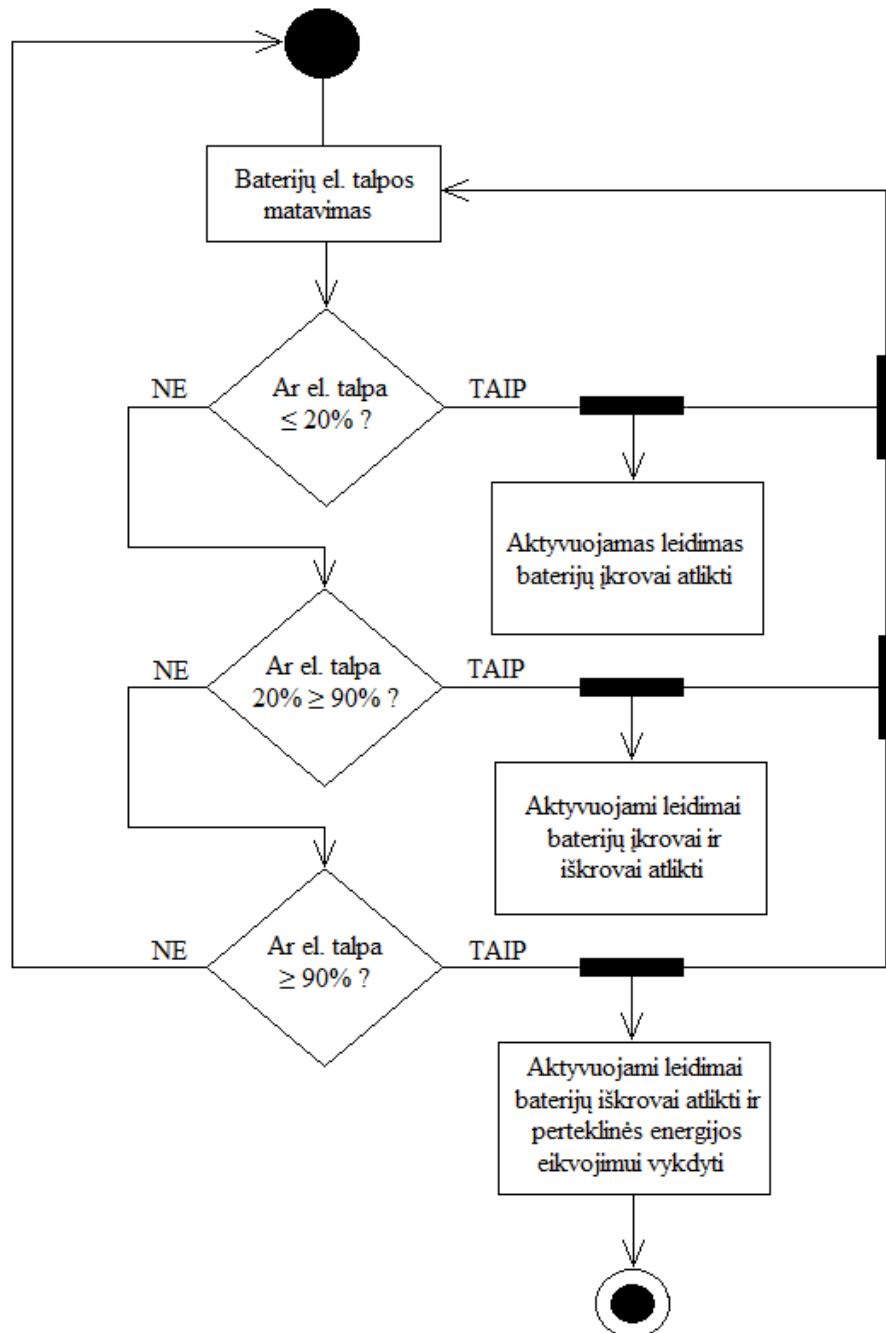
21 pav. VKRĮ schemos būsenos diagrama

Šaltinis: sudaryta autoriaus

Pagal pateiktą VKRĮ būsenos diagramą (žiūr. 21 pav.) matyti, kad pirmiausia yra įvertinamas ar vėjo kryptis pasikeitė, jeigu taip, tada atliekamas vėjo krypties matavimas.

Šiame VKRĮ schemos funkcijos etape, vėjo krypties matavimo prietaiso analoginis signalas (0...5V) ADC keitiklio pagalba konvertuojamas į skaitmeninį 16 bitų signalą. Pagal skaitmeninio signalo reikšmę (trečiajame funkcijos etape) atliekamas vėjo jėgainės kampo reguliavimas tarp oro srauto krypties ir horizontaliosios ašies (vėjo krypties matavimas nenutraukiamas).

BKĮ (Baterijos Krovos Įtaiso) schemos pagrindinės funkcijos – matuoti baterijos įkrovos/iškrovos reikšmes bei vykdyti baterijos įkrovos/iškrovos kontrolę. Pirmajai funkcijai atlikti naudojamas ADC keitiklis, kuris konvertuoja analoginį signalą į 16 bitų skaitmeninį signalą.



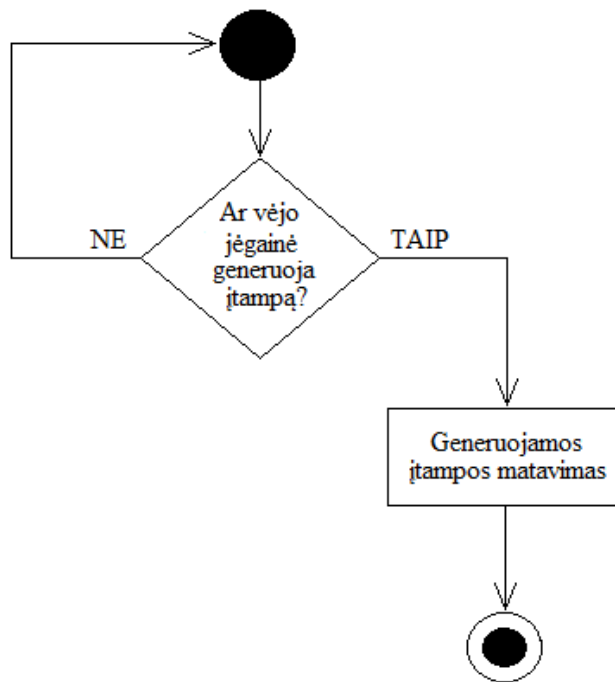
22 pav. BKĮ schemos būsenos diagrama

Šaltinis: sudaryta autoriaus

Skaitmeninis (aukšto ir žemo fronto) signalas perduodamas FPGA matricoje projektuojamai BKĮ schemai (vėjo jėgainės valdymo sistemos Nr.: 1 realizavimo atveju) arba Atmega103 mikrovaldiklio architektūros elementams, kurie įgyvendinti FPGA matricoje (vėjo jėgainės valdymo sistemos Nr.: 2 realizavimo atveju). BKĮ schemeje realizuotas ADIA (Analog Digital Interface Adapter) elementas, kuris geba priimti ADC siunčiamus aukšto ir žemo fronto signalus. Sekančiai funkcijai atlikti naudojamas algoritmas, pagal kurį atliekamas baterijos įkrovos/iškrovos valdymas, siekiant apsaugoti bateriją nuo giles el. energijos iškrovos arba perkrovos.

Pagal pateiktą BKĮ būsenos diagramą (žiūr. 22 pav.) matyti, kad pirmiausia atliekamas baterijų bloko el. talpos matavimas. Esant apkrovai, baterijų talpą galima matuoti pagal įtampos dydį, taigi analoginė įtampos reikšmė yra konvertuojama į skaitmeninę reikšmę ADC keitiklio pagalba. Tokiu būdu atliekama el. talpos matavimas baterijų blokuose. Būsenų diagramoje matyti, kad esant tam tikrai baterijų įkrovai aktyvuojami leidimai (įkrovos, iškrovos bei perteklinės energijos eikvojimo). Taigi, jeigu baterijų blokas yra įkrautas **ne daugiau 20 proc.** tada aktyvuojamas įkrovos leidimas (t.y. baterija pradeda kaupti, vėjo jėgainei generuojant el. energiją) bei uždraudžiama atlikti iškrovos bei perteklinės energijos eikvojimo procedūras. Esant baterijų blokui įkrautam **ne mažiau 20 proc. ir ne daugiau 90 proc.**, aktyvuojami el. energijos įkrovos ir iškrovos leidimai (t.y. vėjo jėgainės generuojama el. energija yra nukreipiama į baterijų bloką bei tuo pačiu metu, esant el. energijos vartotojo poreikiui, iškraunama baterijoje sukaupta el. energija) bei uždraudžiama perteklinę el. energiją eikvoti veltui. Jeigu baterijų blokas įkrautas **ne mažiau 90 proc.** tada suteikiami leidimai eikvoti perteklinę el. energiją bei iškrauti baterijų bloką, atitinkamai uždraudžiamas įkrovos vykdymas. Priskiriant leidimus, pagal šį valdymo algoritmą, baterijų bloko el. talpos matavimas yra atliekamas nenutrūkstamai.

GĮMĮ (Generuojamos Įtampos Matavimo Įtaiso) schemos pagrindinė funkcija – matuoti generuojamą įtampą. Šiai funkcijai atlikti naudojamas ADC keitiklis, kuris konvertuoja analoginį signalą į 16 bitų skaitmeninį. Skaitmeninis (aukšto ir žemo fronto) signalas perduodamas FPGA matricoje projektuojamai GĮMĮ schemai (vėjo jėgainės valdymo sistemos Nr.: 1 realizavimo atveju) arba Atmega103 mikrovaldiklio architektūros elementams, kurie įgyvendinti FPGA matricoje (vėjo jėgainės valdymo sistemos Nr.: 2 realizavimo atveju). GĮMĮ schemeje realizuotas ADIA (Analog Digital Interface Adapter) elementas, kuris geba priimti ADC siunčiamus aukšto ir žemo fronto signalus. Pagal pateiktą GĮMĮ būsenos diagramą (žiūr. 23 pav.) matyti, kad generuojamos įtampos matavimas atliekamas pradėjus vėjo jėgainei generuoti el. energiją. Taigi nepertraukiamu režimu yra tikrinama ar vėjo jėgainė pradėjo generuoti įtampą, jeigu taip tada atliekamas tam tikro dydžio analoginio signalo matavimas bei konvertavimas į skaitmeninį signalą, kuris nukreipiamas į valdymo sistemos centrinį vykdymo įtaisą.



23 pav. GIMI schemos būsenos diagrama

Šaltinis: sudaryta autoriaus

UART_TX įtaiso funkcija – perduoti duomenys vartotojui į PC (personalinį kompiuterį), naudojant UART protokolą, taigi šiai funkcijai atlikti sukuriamas TX (angl. „Transmit“) algoritmas. Schemoje realizuotas buferis, kuriame talpinami visų išnagrinėtų schemų momentinės reikšmės. Esant užpildytam buferiui, informaciniai bitai yra išstumiami per UART_TX įtaiso SER_OUT išėjimo signalą į personalinio kompiuterio RX (angl. „Receive“) schemą. Žemiau esančiame paveikslėlyje pateiktas UART_TX buferis, kuriame matyti visų anksčiau aptartų schemų informaciniai bitai:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39		
"Start" bitas	VGMĮ bitas [0]	VGMĮ bitas [1]	VGMĮ bitas [2]	VGMĮ bitas [3]	VGMĮ bitas [4]	VGMĮ bitas [5]	VGMĮ bitas [6]	VGMĮ bitas [7]	VGMĮ bitas [8]	VGMĮ bitas [9]	VGMĮ bitas [10]	VGMĮ bitas [11]	VGMĮ bitas [12]	VGMĮ bitas [13]	VGMĮ bitas [14]	VGMĮ bitas [15]	VGMĮ bitas [16]	VGMĮ bitas [17]	VGMĮ bitas [18]	VGMĮ bitas [19]	VGMĮ bitas [20]	VGMĮ bitas [21]	VGMĮ bitas [22]	VGMĮ bitas [23]	VGMĮ bitas [24]	VGMĮ bitas [25]	VGMĮ bitas [26]	VGMĮ bitas [27]	VGMĮ bitas [28]	VGMĮ bitas [29]	VGMĮ bitas [30]	VGMĮ bitas [31]	VGMĮ bitas [32]	VGMĮ bitas [33]	VGMĮ bitas [34]	VGMĮ bitas [35]	VGMĮ bitas [36]	VGMĮ bitas [37]	VGMĮ bitas [38]	VGMĮ bitas [39]	"Stop" bitas

24 pav. UART_TX schemos informaciniai bitai

Šaltinis: sudaryta autoriaus

Visi 24 paveikslėlyje pavaizduoti informaciniai bitai RX schemai perduodami postūmiu per SER_OUT vieno bito išėjimo signalą. Personalinio kompiuterio RX schemai aptikus „start“ ir „stop“ bitus laikoma, kad informacija tinkama perduota bei galima atlikti šios informacijos pateikiama vartotojui.

Vėjo jėgainių valdymo sistemos Nr.: 1 atveju, realizuojant valdymo algoritmą (kuris susideda iš išvardintų schemų bei jų vykdymo funkcijų) FPGA matricoje sudaromi loginiai elementai bei ryšiai

tarp jų (apie valdymo algoritmo įgyvendinimą FPGA matricoje plačiau aprašyta skyriuje „FPGA matricos architektūros apžvalga“). Priklausomai nuo programuojamų funkcijų, naudojamas skirtingas loginių elementų (loginių celių, dalelių, trigerių ir t.t.) bei ryšio linijų skaičius, tipas, ilgis, išdėstymas. Sekančios valdymo sistemos realizacijos atveju, t.y. įgyvendinant algoritmą vėjo jėgainių valdymo sistemoje Nr.: 2, naudojami pastovūs (nekintantys) funkciniai elementai (nepriklausomai nuo valdymo algoritmo sudėtingumo ar funkcionalumo), tačiau algoritmas talpinamas (per)programuojamoje programinėje atmintyje, todėl priklausomai nuo valdymo algoritmo sudėtingumo, išnaudojamas skirtingas atminties dydis (Atmega103 mikrovaldiklio maks. programinė atmintis yra 128KB).

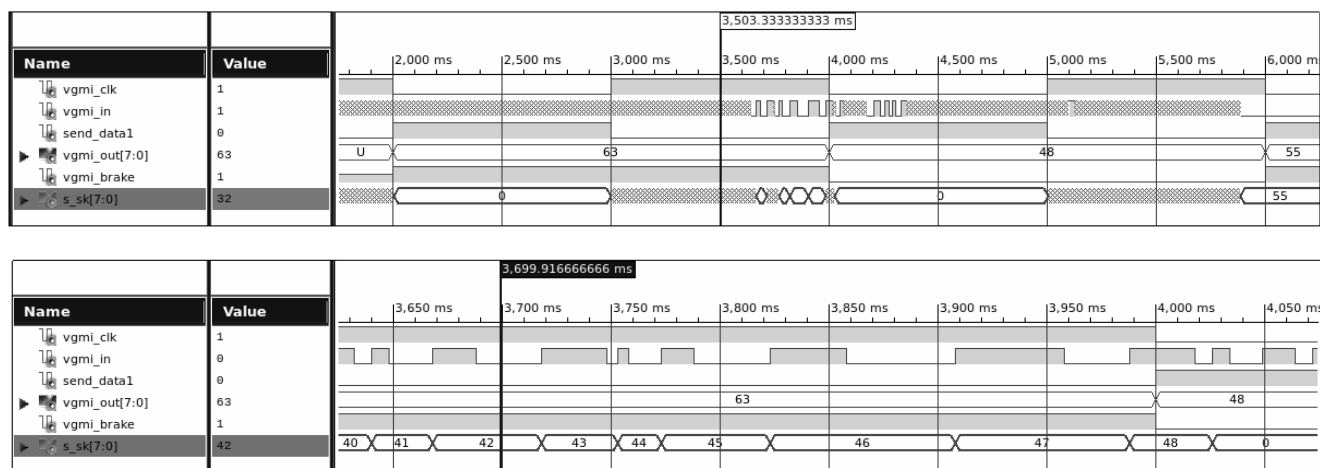
2.2.3 Tiriamojo darbo realizacijų analizavimas

Vėjo jėgainės valdymo sistemų (Nr.: 1 ir Nr.: 2) algoritmas sudarytas iš pagrindinių schemų (VGMĮ, VKRĮ, BKĮ, GĮMĮ ir UART_TX), kurios atlieka skirtingas funkcijas. Šių schemų funkcionalumas aprašytas viršuje esančiame poskyriuje „Tiriamojo darbo metodai“, taigi šioje tiriamojo darbo dalyje pateikti realizacijos rezultatai pagal aprašytas schemų funkcijas. Siekiant įgyvendinti skirtingas valdymo sistemas su vienodais valdymo algoritmais ir tarpusavyje jas palyginti, pirmiausia atliekamas valdymo algoritmo modeliavimas. Modeliavimo metu analizuojama ir testuojama schema bei jos funkcionalumas, toliau norint valdymo algoritmą įgyvendinti FPGA matricoje būtina atlikti algoritmo sintezę, naudojant konkrečios FPGA matricos bibliotekas (šiuo atveju Spartan – 6). Žemiau esančiuose poskyriuose pateikti bei aprašyti vėjo elektrinių valdymo sistemų modeliavimo ir sintezavimo rezultatai.

2.2.3.1 Vėjo jėgainės valdymo sistemos Nr.: 1 realizacija

VGMĮ schemos modeliavimo rezultatai pavaizduoti 25 paveikslėlyje, pagal kurį matyti, kad VGMĮ schemą sudaro pagrindiniai šeši signalai (t.y. „vgmi_clk“, „vgmi_in“, „send_data1“, „vgmi_out“, „vgmi_brake“, „s_sk“). Pirmasis „vgmi_clk“ – tai sinchroninio dažnio įėjties signalas, kurio periodas yra 2 sekundės (pusperiodis – 1 sekundė). Šio signalo pagalba, anemometro generuojamo impulsinio signalo skaitikliui, nustatomas pradžios ir pabaigos intervalas. Taigi esant „vgmi_clk“ aukšto fronto signalui atliekamas impulsinių signalų skaičiavimas (vienos sekundės intervale), kai „vgmi_clk“ pereina į žemą frontą tada atliekamas VGMĮ schemos rezultatų perdavimas. Sekantis „vgmi_in“ – tai anemometro generuojamas „reed“ kontakto impulsinis įėjties signalas. Trečiasis „send_data“ – VGMĮ įtaiso darbo būsenos identifikavimo išėjties signalas. Esant „send_data1“ aukštam frontui, informuojama, kad impulsiniai signalai yra suskaičiuoti bei atliekamas rezultatų perdavimas, jeigu šis signalas pereina į žemą frontą, tada suteikiama informacija, kad yra atliekamas impulsinių signalų skaičiavimas. Sekantis „Vgmi_out“ – tai rezultatų išėjties signalas, kuris

perduoda vartotojui (arba sekančiai FPGA matricoje realizuotai schemai) suskaičiuotą rezultatą (momentinį vėjo greitį, išreikštu impulsų pasikartojimo dažniu [Hz]). Priešpaskutinis „vgmi_brake“ – tai rotoriaus apsisukimų apribojimo išėjties signalas, kuris pereina į aukštą frontą, jeigu pasiekiamas kritinis vėjo greitis (šiuo atveju 25 m/s). Paskutinis modeliavimo rezultatuose pateiktas elementas „s_sk“ – tai skaitliuko 8 bitų buferis, kuriame kaupiami sumuojami „vgmi_in“ impulsiniai signalai.

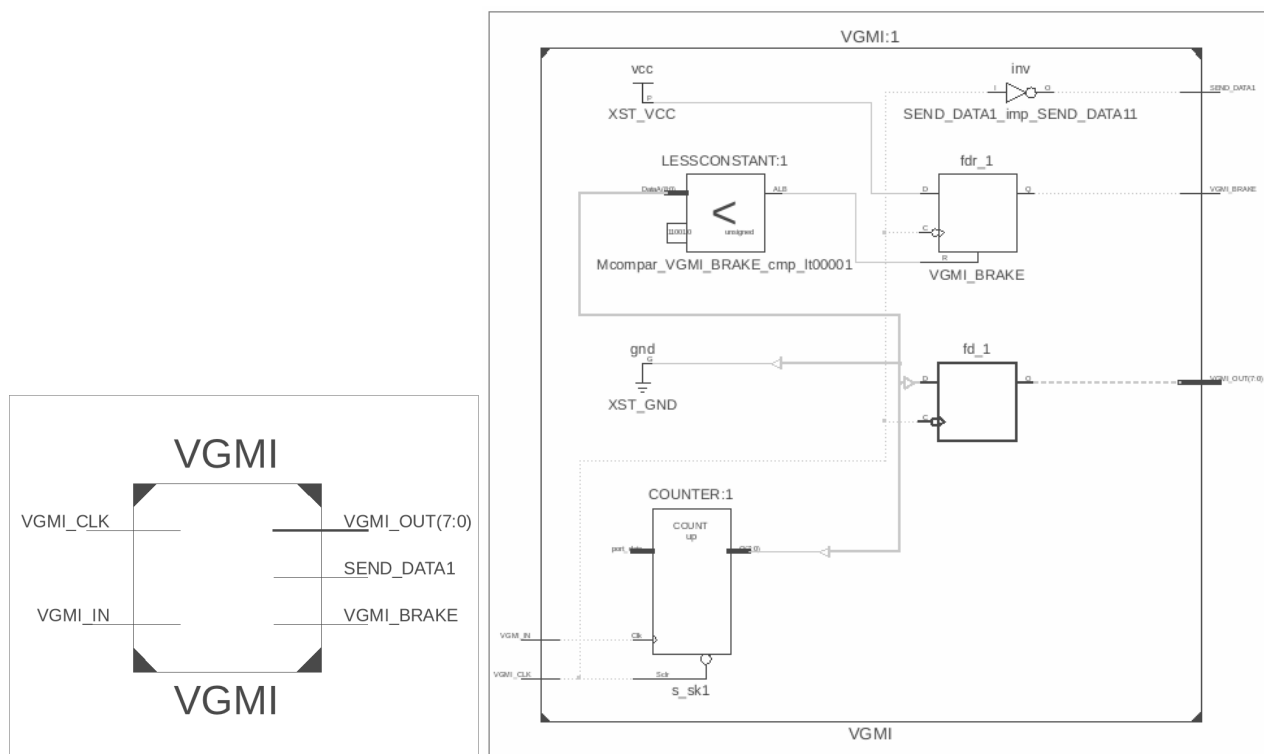


25 pav. VGMi schemos modeliavimo rezultatai

Šaltinis: sudaryta autoriaus

Antroje 25 paveikslėlio dalyje (žemiau esančiame modeliavimo rezultatų atvaizde), matyti padidintą VGMi modeliavimo rezultatų vaizdą, kuriame paryškintas skaitiklio 8 bitų buferis „s_sk“. Buferyje kaupiami sumuojami anemometro generuojami impulsai tol, kol „vgmi_clk“ yra aukšto fronto signalas, perėjus „vgmi_clk“ į žemą frontą – paskutinė „s_sk“ esanti reikšmė perduodama į „vgmi_out“ aštuonių bitų išėjties signalą, o „s_sk“ buferis išvalomas.

Siekiant šią schemą įgyvendinti FPGA matricoje, atlikę modeliavimą bei gavę tikėtiną VGMi schemos procedūrų funkcionalumą, būtina atlikti sintezę. Žemiau esančiame 26 paveikslėlyje matyti VGMi schemos sintezavimo rezultatai. Paveikslėlio kairėje pusėje pateiktas bendras sintezuotos VGMi schemos bloko vaizdas, kuris sudarytas iš vidinių loginių elementų. Šio bloko kairėje pusėje pavaizduoti anksčiau aptarti VGMi schemos išoriniai įėjčių signalai. Atitinkamai, bloko dešinėje pusėje matyti – trys VGMi schemos išėjčių signalus. Toliau 26 paveikslėlio dešinėje pusėje pavaizduotas sintezuoto VGMi bloko išskleistas vaizdas. Taigi, VGMi blokas sudarytas iš vidinių loginių elementų bei ryšių tarp jų. Šių funkcinių loginių elementų (skaitiklio, buferių, loginių ventilių ir t.t.) visuma ir ryšiai (kuriais sujungti funkciniai elementai vienas su kitu) sudaro bendrą VGMi schemos funkciją.



26 pav. VGMI schemos sintezavimo rezultatai

Šaltinis: sudaryta autoriaus

VKRI schemos modeliavimo rezultatai pavaizduoti 27 paveikslėlyje, kuriame matyti trylika skirtingų signalų, iš kurių trys yra schemos įeities signalai („clk“, „reset“ ir „vkri_sdata“), sekantys šeši schemos išeities signalai („vkri_out“, „cs“, „q1“, „q2“, „q3“, „q4“), vienas buferis („buf“) ir skaitiklis („bit_cnt“) bei du signalai esamos ir sekančios schemos būsenos nustatymui / identifikavimui („current_state“, „next_state“). Taigi „clk“ yra sinchroninio dažnio signalas, kurio taktiniu dažniu (šiuo atveju 4MHz) schema vykdo pavestas funkcijas.

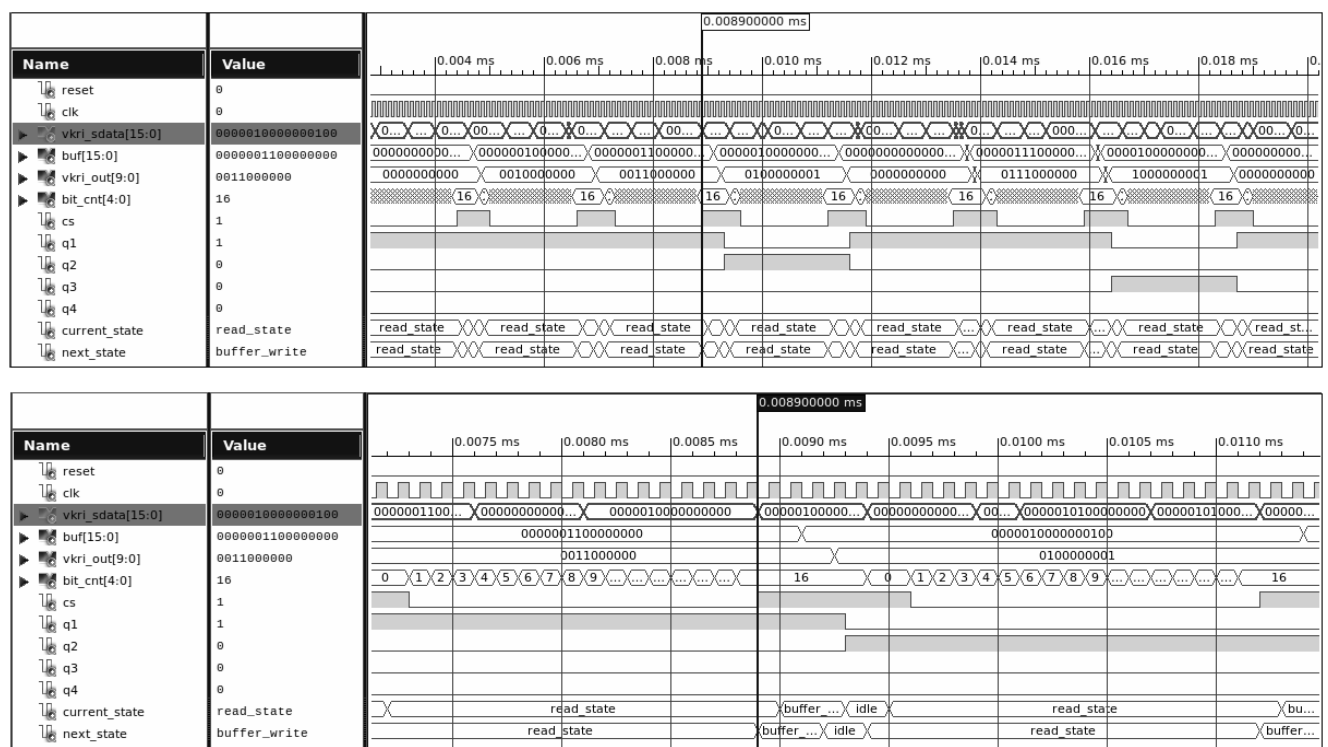
Sekantis „reset“ signalas, skirtas perkrauti VKRI schemą, tokiu būdu išvalomi visi schemos buferiai ir darbas pradamas iš naujo. Siekiant užtikrinti, kad VKRI schema nuo pradžios atliktų jai paskirtą funkcionalumą, todėl dažniausiai šis signalas pradamas vykdyti (t.y. vykdomas perėjimas į aukštą signalo frontą) vos tik startavus schemai. Šiuo atveju po 400 ns „reset“ signalas gražinamas į žemą frontą, tada startuoja VKRI schemos funkcijos.

Toliau nagrinėjant modeliavimo rezultatus, galima aptarti VKRI schemos „vkri_sdata“ įeities signalą. Tai analoginio signalo konvertuota 16 bitų skaitmeninė reikšmė (šis signalas yra siunčiamas iš ADC keitiklio į FPGA matricą). Pagal „vkri_sdata“ reikšmę yra kontroliuojami schemos išėjimo signalai, tokiu būdu valdomas žingsninis variklis, kuris kontroliuoja vėjo jėgainės kampa tarp horizontaliosios turbino ašies ir oro srauto krypties. Siekiant identifikuoti „vkri_sdata“ skaitmeninio signalo esamą bitų būseną, naudojamas „bit_cnt“ skaitiklis, kuris seka ir skaičiuoja kiekvieną ateinantį bitą. Suskaičiavus 16 žingsnių laikoma, kad paskutinė informacija (ties 16 – uoju žingsniu) yra

galutinė ir atitinkanti analoginio signalo vertę. To pasekoje generuojamas aukšto fronto „cs“ išeities signalas, kuris perduodamas ADC keitikliui. Šio signalo pagalba („cs“) ADC keitiklis baigia darbą, tada signalo „vkri_sdata“ reikšmė perduodama į vidinį VKRĮ schemos buferį, iš kurio paimamas rezultatas. Rezultatas perduodamas vartotojui (arba sekančiai FPGA matricos schemai, „vkri_out“ išeities signalo pagalba) bei atliekamas žingsninio variklio valdymas atitinkamai nustatant išeities signalų „q1“, „q2“, „q3“ ir „q4“ aukštus ir žemus frontus.

Atitinkamai, prieš pradėdant „bit_cnt“ skaičiuoti iš naujo – „cs“ išeities signalas gražinamas į žemą frontą, to pasekoje leidamas ADC keitikliui pradėti darbą (momentinę analoginio signalo vertę konvertuoti į skaitmeninį signalą). Taip pradėdamas vykdyti sekančios reikšmės identifikavimas.

Modeliavimo rezultatuose (žiūr. 27 pav.) pavaizduoti „current_state“ ir „next_state“ yra vidiniai signalai, kurie naudojami VKRĮ funkcijų priskyrimui bei schemos valdymui, taip pat jų pagalba identifikuojama esamos ir sekančios būsenų funkcijos. Antroje 27 paveikslėlio dalyje (žemiau esančiame modeliavimo rezultatų atvaizde), matyti padidintą VKRĮ modeliavimo rezultatų vaizdą.

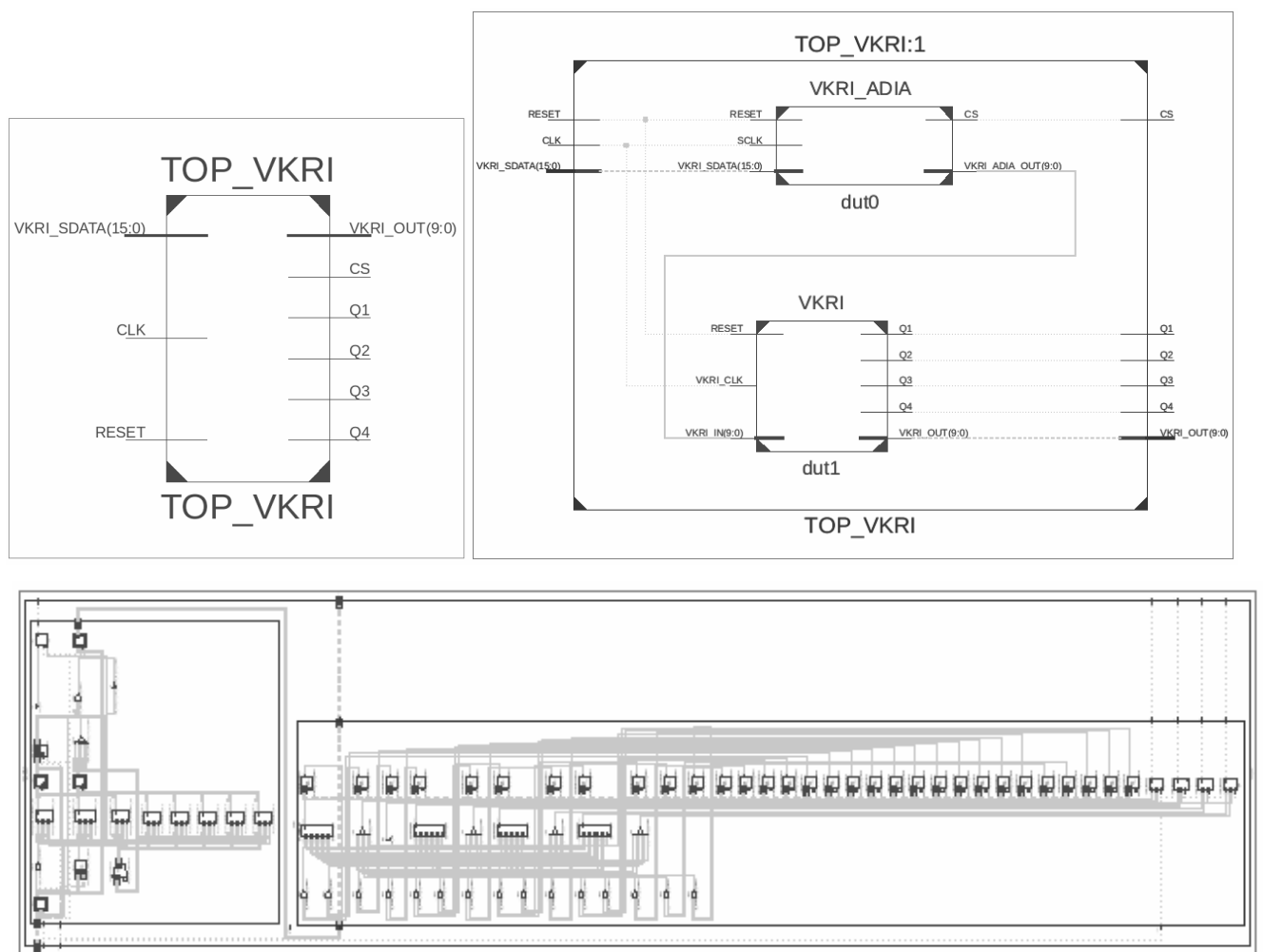


27 pav. VKRĮ schemos modeliavimo rezultatai

Šaltinis: sudaryta autoriaus

Atlikę modeliavimą bei gavę tikėtiną VKRĮ schemos procedūrų funkcionalumą, būtina atlikti sintezę. Žemiau esančiame 28 paveikslėlyje matyti VKRĮ schemos sintezavimo rezultatai. Paveikslėlio kairėje pusėje pateiktas bendras sintezuotos VKRĮ schemos bloko vaizdas, kuris sudarytas iš vidinių loginių elementų. Šio bloko kairėje pusėje pavaizduoti anksčiau aptarti VKRĮ schemos išoriniai įėjimų

signalai. Atitinkamai, bloko dešinėje pusėje matyti – trys VKRĮ schemos išeičių signalus. Toliau 28 paveikslėlio dešinėje pusėje pavaizduotas sintezuoto VKRĮ bloko išskleistas vaizdas. Taigi, VKRĮ blokas sudarytas iš pagrindinių vidinių loginių elementų (VKRI_ADIA – dut0 ir VKRI – dut1) bei ryšių tarp jų. VKRI_ADIA funkcinis elementas skirtas priimti skaitmeninį signalą iš ADC keitiklio, jį apdoroti ir perduoti VKRĮ funkciniam elementui. Šis funkcinis elementas (blokas) atlieka žingsninio variklio valdymo bei informacijos pateikimo vartotojui (arba sekančiai FPGA matricos schemai) funkcijas. VKRI_ADIA ir VKRI funkciniai blokai sudaryti iš įvairių loginių elementų (skaitiklių, buferių, triggerių, loginių ventilių ir t.t.) visumos ir ryšių (žiūr. 28 pav. apatinį sintezavimo rezultatų atvaizdą).



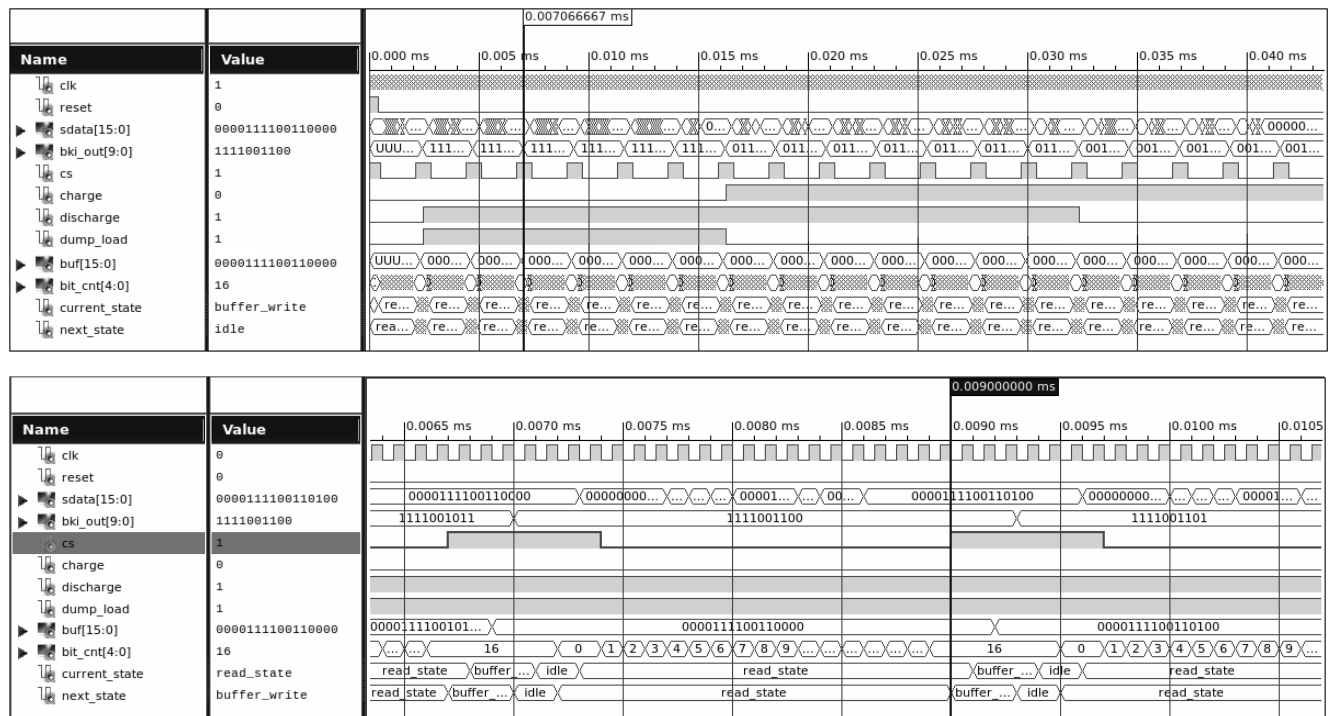
28 pav. VKRĮ schemos sintezavimo rezultatai

Šaltinis: sudaryta autoriaus

BKĮ schemos modeliavimo rezultatai pavaizduoti 29 paveikslėlyje, kuriame matyti dvylika skirtingų signalų, iš kurių trys yra schemos įeities signalai („clk“, „reset“ ir „sdata“), sekantys penki schemos išeities signalai („bki_out“, „cs“, „charge“, „discharge“, „dump_load“), vienas buferis („buf“) ir skaitiklis („bit_cnt“) bei du signalai esamos ir sekančios schemos būsenos nustatymui /

identifikavimui („current_state“, „next_state“). Taigi „clk“ yra sinchroninio dažnio signalas, kurio taktiniu dažniu (šiuo atveju 4MHz) schema vykdo pavestas funkcijas.

Sekantis „reset“ signalas, skirtas perkrauti BKĮ schemą, tokiu būdu išvalomi visi schemos buferiai ir darbas pradedamas iš naujo. Siekiant užtikrinti, kad BKĮ schema nuo pradžios atliktų jai paskirtą funkcionalumą, todėl dažniausiai šis signalas pradedamas vykdyti (t.y. vykdomas perėjimas į aukštą signalo frontą) vos tik startavus schemai. Šiuo atveju po 400 ns „reset“ signalas gražinamas į žemą frontą, tada startuoja BKĮ schemos funkcijos.



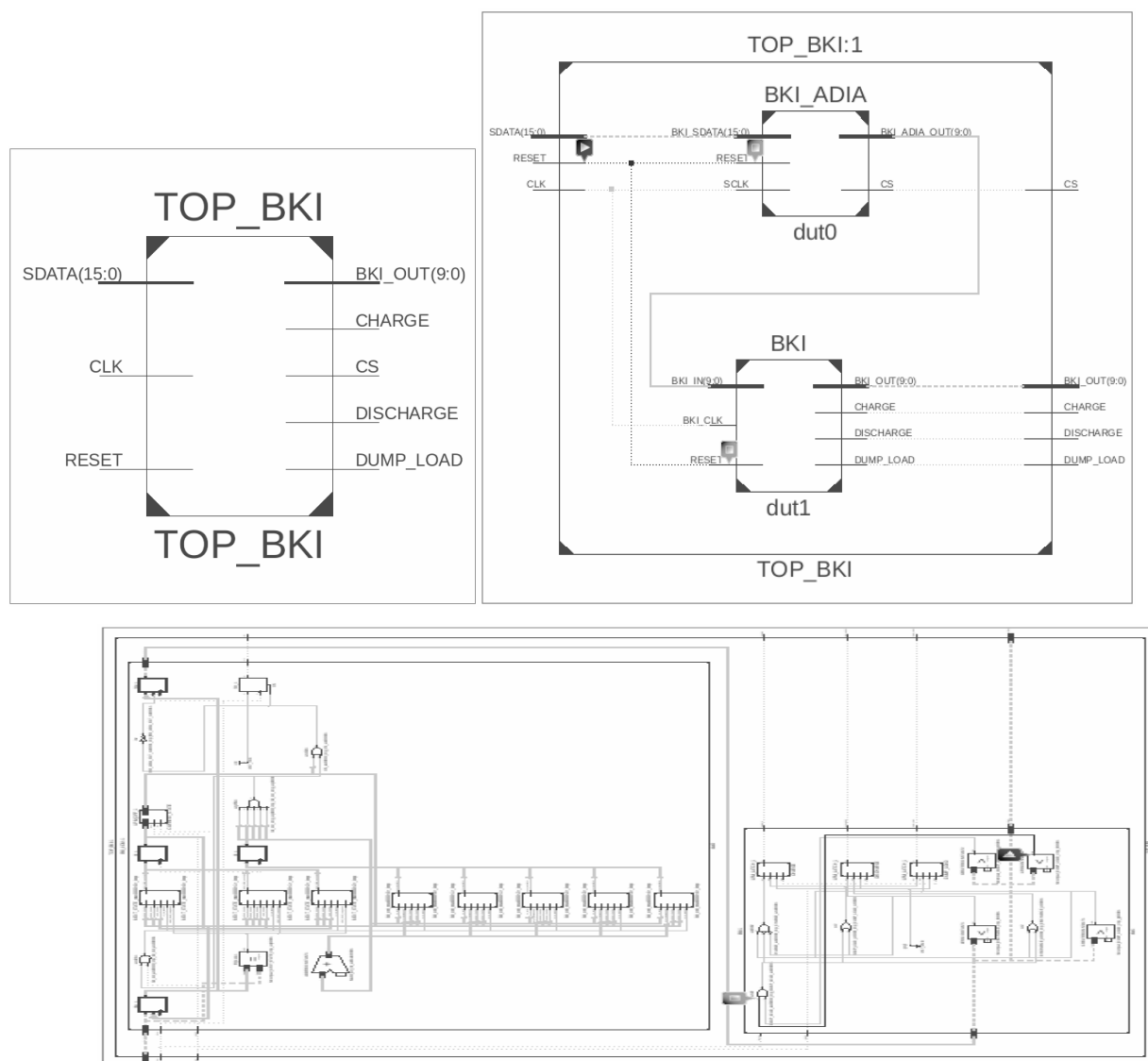
29 pav. BKĮ schemos modeliavimo rezultatai

Šaltinis: sudaryta autoriaus

BKĮ schemos „sdata“ įeities signalas – tai analoginio signalo konvertuota 16 bitų skaitmeninė reikšmė (šis signalas yra siunčiamas iš ADC keitiklio į FPGA matricą). Pagal „sdata“ reikšmę yra kontroliuojami schemos išėjimo signalai, tokiu būdu tinkamai valdomas baterijų bloko įkrovimas / iškrovimas. Siekiant identifikuoti „sdata“ skaitmeninio signalo esamą bitų būseną, naudojamas „bit_cnt“ skaitiklis, kuris seka ir skaičiuoja kiekvieną ateinantį bitą. Suskaičiavus 16 žingsnių laikoma, kad paskutinė informacija (ties 16 – uoju žingsniu) yra galutinė ir atitinkanti analoginio signalo vertę. To pasekoje generuojamas aukšto fronto „cs“ išeities signalas, kuris perduodamas ADC keitikliui. Šio signalo pagalba („cs“) ADC keitiklis baigia darbą, tada signalo „sdata“ reikšmė perduodama į vidinį BKĮ schemos buferį, iš kurio paimamas rezultatas. Rezultatas perduodamas vartotojui (arba sekančiai FPGA matricos schemai, „bki_out“ išeities signalo pagalba) bei atliekamas baterijų bloko įkrovos / iškrovos tinkamas valdymas, (de)aktyvuojant leidimus: baterijų bloko krovos – „charge“, baterijų

bloko iškrovos – „discharge“ bei perteklinės energijos eikvojimo – „dump_load“. Atitinkamai, prieš pradėdant „bit_cnt“ skaičiuoti iš naujo – „cs“ išeities signalas gražinamas į žemą frontą, to pasekoje leisdamas ADC keitikliui pradėti darbą (momentinę analoginio signalo vertę konvertuoti į skaitmeninį signalą). Taip pradėdamas vykdyti sekančios reikšmės identifikavimas.

Modeliavimo rezultatuose (žiūr. 29 pav.) pavaizduoti „current_state“ ir „next_state“ vidiniai signalai, kurie naudojami BKĮ funkcijų priskyrimui bei schemos valdymui, taip pat jų pagalba identifikuojama esamos ir sekančios būsenų funkcijos. Antroje 29 paveikslėlio dalyje (žemiau esančiame modeliavimo rezultatų atvaizde), matyti padidintą BKĮ modeliavimo rezultatų vaizdą.



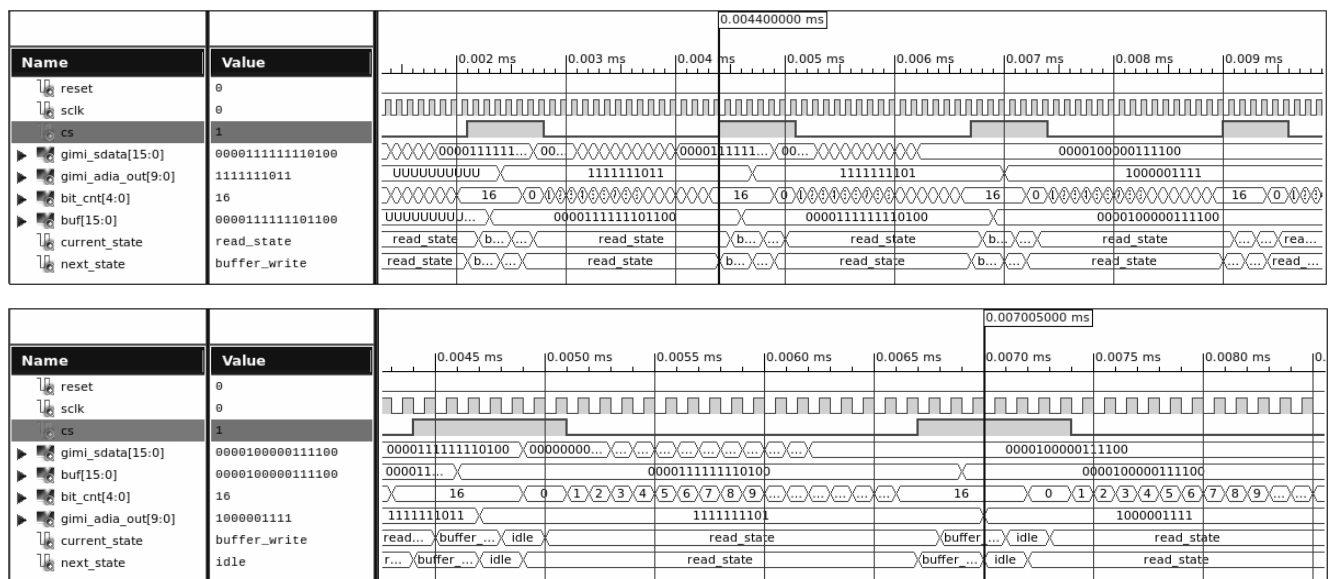
30 pav. BKĮ schemos sintezavimo rezultatai

Šaltinis: sudaryta autoriaus

Atlikę modeliavimą bei gavę tikėtiną BKĮ schemos procedūrų funkcionalumą, būtina atlikti sintezę. Aukščiau esančiame 30 paveikslėlyje matyti BKĮ schemos sintezavimo rezultatai. Paveikslėlio kairėje pusėje pateiktas bendras sintezuotos BKĮ schemos bloko vaizdas, kuris sudarytas iš vidinių

loginių elementų. Šio bloko kairėje pusėje pavaizduoti anksčiau aptarti BKĮ schemos išoriniai įeičių signalai. Atitinkamai, bloko dešinėje pusėje matyti – trys schemos išeičių signalus. Toliau 30 paveikslėlio dešinėje pusėje pavaizduotas sintezuoto BKĮ bloko išskleistas vaizdas. Taigi, BKĮ blokas sudarytas iš pagrindinių vidinių loginių elementų (BKI_ADIA – dut0 ir BKI – dut1) bei ryšių tarp jų. BKI_ADIA funkcinis elementas skirtas priimti skaitmeninį signalą iš ADC keitiklio, jį apdoroti ir perduoti BKI funkciniam elementui. Šis funkcinis elementas (blokas) atlieka baterijų bloko krovos kontrolę bei vykdo informacijos pateikimo vartotojui (arba sekančiai FPGA matricai) funkcijas. BKI_ADIA ir BKI funkciniai blokai sudaryti iš įvairių loginių elementų (skaitiklių, buferių, triggerių, loginių ventilių ir t.t.) visumos ir ryšių tarp jų (žiūr. 30 pav. apatinį sintezavimo rezultatų atvaizdą).

GĮMĮ schemos modeliavimo rezultatai pavaizduoti 31 paveikslėlyje, kuriame matyti devynis skirtingus signalus, iš kurių trys yra schemos įeities signalai („sclk“, „reset“ ir „gimi_sdata“), sekantys du schemos išeities signalai („gimi_adia_out“, „cs“), vienas buferis („buf“) ir skaitiklis („bit_cnt“) bei du signalai esamos ir sekančios schemos būsenos nustatymui / identifikavimui („current_state“, „next_state“). Taigi „sclk“ yra sinchroninio dažnio signalas, kurio taktiniu dažniu (šiuo atveju 4MHz) schema vykdo pavestas funkcijas.



31 pav. GĮMĮ schemos modeliavimo rezultatai

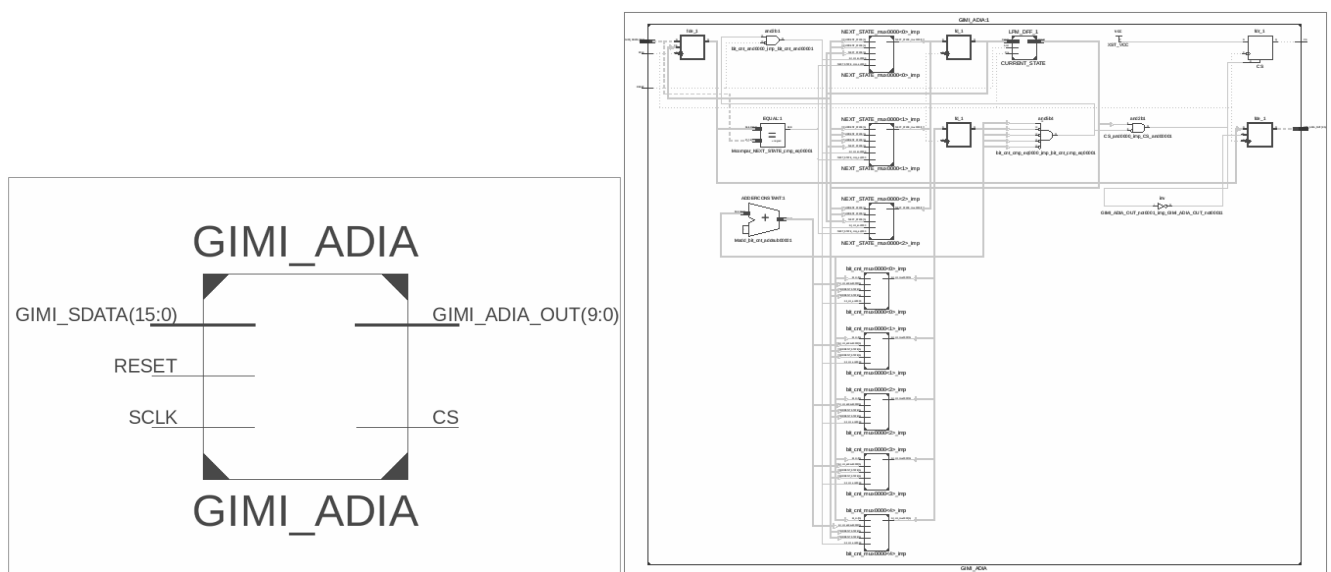
Šaltinis: sudaryta autoriaus

„Reset“ signalas, skirtas perkrauti GĮMĮ schemą, tokiu būdu išvalomi visi schemos buferiai ir darbas pradamas iš naujo. Siekiant užtikrinti, kad GĮMĮ schema nuo pradžios atliktų jai paskirtą funkcionalumą, todėl dažniausiai šis signalas pradamas vykdyti (t.y. vykdomas perėjimas į aukštą signalo frontą) vos tik startavus schemai. Šiuo atveju po 400 ns „reset“ signalas gražinamas į žemą frontą, tada startuoja GĮMĮ schemos funkcijos.

GIMI schemos „gimi_sdata“ įeities signalas – kaip ir aukščiau aprašytų schemų tai analoginio signalo konvertuota 16 bitų skaitmeninė reikšmė (šis signalas yra siunčiamas iš ADC keitiklio į FPGA matricą). Siekiant identifikuoti „gimi_sdata“ skaitmeninio signalo esamą bitų būseną, naudojamas „bit_cnt“ skaitiklis, kuris seka ir skaičiuoja kiekvieną ateinantį bitą. Suskaičiavus 16 žingsnių laikoma, kad paskutinė informacija (ties 16 – uoju žingsniu) yra galutinė ir atitinkanti analoginio signalo (šiuo atveju matuojamos generuojamos įtampos) vertę. To pasekoje perduodamas ADC keitikliui aukšto fronto „cs“ išeities signalas. Šio signalo pagalba („cs“) ADC keitiklis baigia darbą ir signalo „gimi_sdata“ reikšmė perduodama į vidinį GIMI schemos buferį, iš kurio paimamas rezultatas. Rezultatas perduodamas vartotojui (arba sekančiai FPGA matricai „bki_out“ išeities signalo pagalba).

Atitinkamai, prieš pradėdant „bit_cnt“ skaičiuoti iš naujo – „cs“ išeities signalas gražinamas į žemą frontą, to pasekoje leidamas ADC keitikliui pradėti darbą (momentinę analoginio signalo vertę konvertuoti į skaitmeninį signalą). Taip pradėdamas vykdyti sekančios reikšmės identifikavimas.

Modeliavimo rezultatuose (žiūr. 31 pav.) pavaizduoti „current_state“ ir „next_state“ vidiniai signalai, kurie naudojami GIMI funkcijų priskyrimui bei schemos valdymui, taip pat jų pagalba identifikuojama esamos ir sekančios būsenų funkcijos. Antroje 31 paveikslėlio dalyje (žemiau esančiame modeliavimo rezultatų atvaizde), matyti padidintą GIMI modeliavimo rezultatų vaizdą.



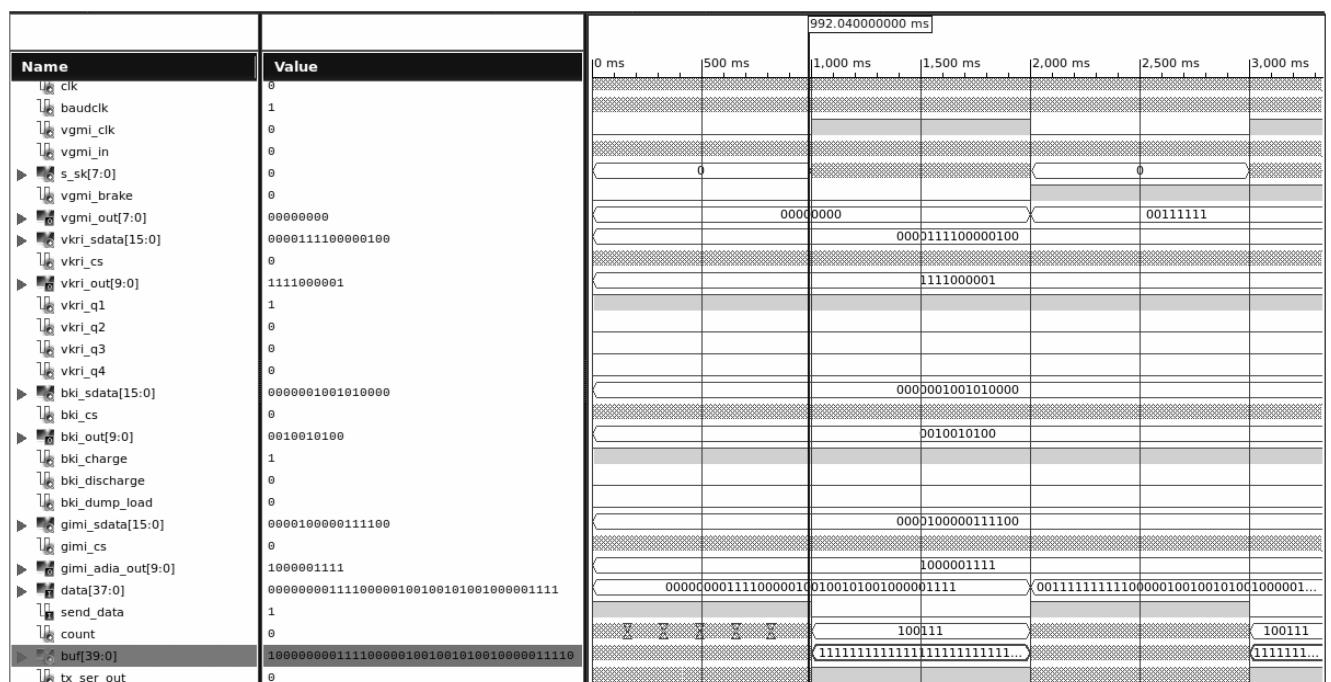
32 pav. GIMI schemos sintezavimo rezultatai

Šaltinis: sudaryta autoriaus

Atlikę modeliavimą bei gavę tikėtiną GIMI schemos procedūrų funkcionalumą, būtina atlikti sintezę. Aukščiau esančiame 32 paveikslėlyje matyti GIMI schemos sintezavimo rezultatai. Paveikslėlio kairėje pusėje pateiktas bendras sintezuotos schemos bloko vaizdas, kuris sudarytas iš vidinių loginių elementų. Šio bloko kairėje pusėje pavaizduoti anksčiau aptarti GIMI schemos išoriniai įeities signalai. Atitinkamai, bloko dešinėje pusėje matyti – trys schemos išeičių signalai. Toliau 32

paveikslėlio dešinėje pusėje pavaizduotas sintezuoto GIMI bloko išskleistas vaizdas. Taigi, GIMI blokas sudarytas iš vieno loginio elemento (GIMI_ADIA). Šis elementas skirtas priimti skaitmeninį signalą iš ADC keitiklio, jį apdoroti ir perduoti vartotojui arba sekančiai FPGA realizuotai schemai. GIMI_ADIA funkciniai blokai sudaryti iš įvairių loginių elementų (skaitiklių, buferių, trigerių, loginių ventilių ir t.t.) visumos ir ryšių tarp jų (žiūr. 32 pav. apatinį sintezavimo rezultatų atvaizdą).

Atlikus visų schemų modeliavimą, testavimą bei sintezavimą, galima visus blokus (VGMI, VKRI, BKI, GIMI bei UART_TX) sujungti į vieną pagrindinį bloką TOP. Tokiu būdu TOP schema turės galimybę atlikti visų šalutinių blokų funkcijas. Pakartotinai atlikus TOP schemai modeliavimą, testavimą bei sintezavimą, schema įgyvendinama FPGA matricoje.



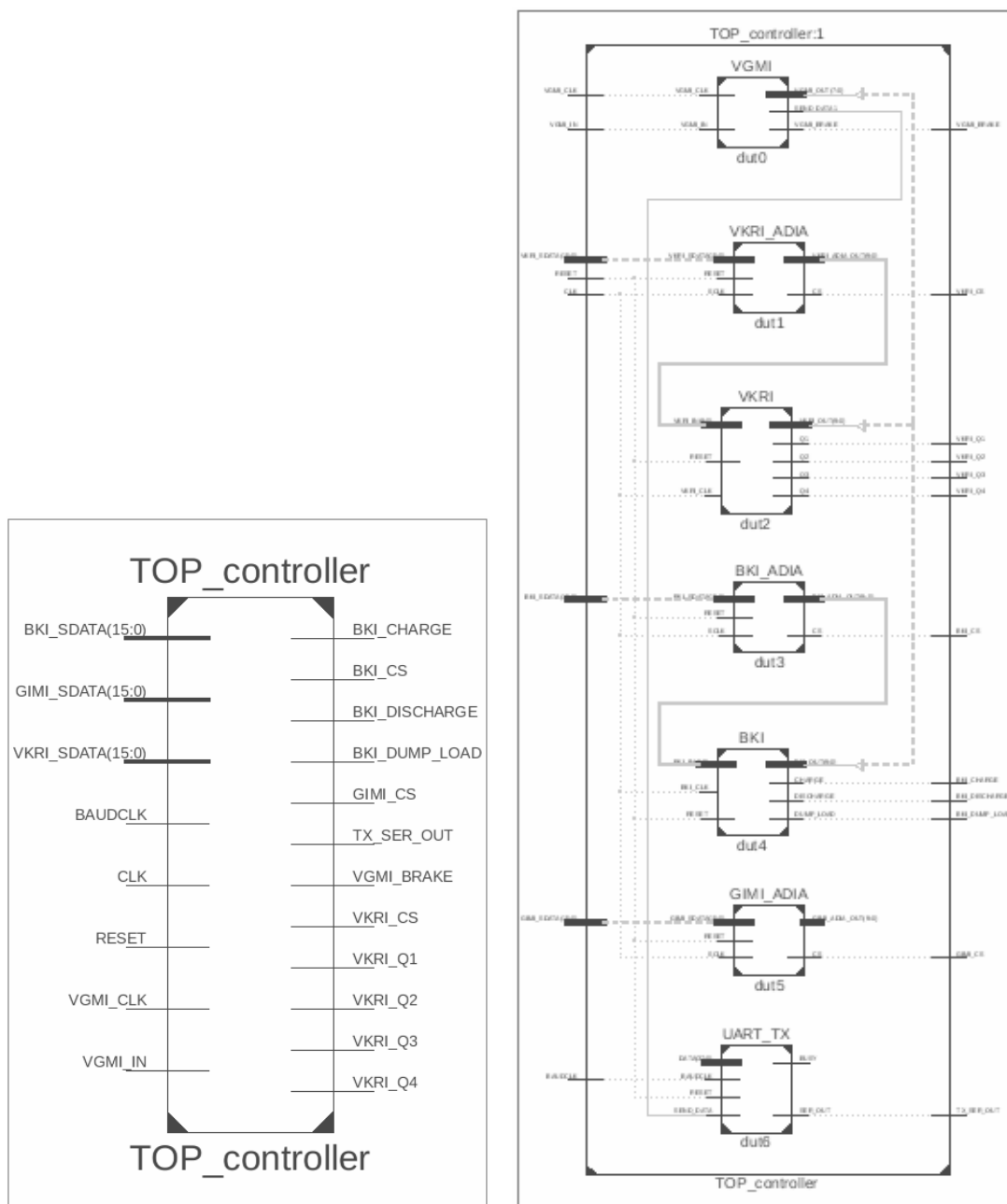
33 pav. TOP schemos modeliavimo rezultatai

Šaltinis: sudaryta autoriaus

Esančiame TOP schemos modeliavimo rezultatų paveikslėlyje (žiūr. 33 pav.) matyti visi anksčiau išvardinti schemų (VGMI, VKRI, BKI ir GIMI) signalai bei jų kitimas laiko intervale. Taip pat galima pastebėti keletą nepamintų signalų, tai UART_TX schemos įeičių, išeičių bei kitų funkcinų elementų (skaitiklio, buferio) signalai. Taigi nagrinėjant 33 paveikslėlio modeliavimo rezultatus, galima teigti, kad UART_TX schemos įeičių signalai yra „clk“, „baudclk“, reset ir „send_data“, išeičių signalas – „ser_out“ bei vidinis 40 bitų buferis „buf“ ir skaitiklis „count“.

UART_TX schemos įeitys „clk“ ir „baudclk“ – tai sinchroninio dažnio signalai, kur „clk“ yra bendras visų schemų taktinio dažnio signalas (4Mhz), kuris skirtas procedūroms vykdyti, o „baudclk“ – tai UART_TX signalas, kurio pagalba perduodama informacija vartotojui (pvz. personalinio kompiuterio RX schemai). Šio signalo taktiniu dažniu yra vykdomas UART protokolas.

„Reset“ signalas, naudojamas vienas visoms schemoms (tuo tarpu ir UART_TX). Kaip aptarta anksčiau, šis signalas skirtas schemų perkrovai, t.y. perkrovus schemas išvalomi visi buferiai ir darbas pradedamas iš naujo. Siekiant užtikrinti, kad schemų junginiai (t.y. pagrindinė TOP schema) nuo pradžios atliktų jai paskirtą funkcionalumą, todėl dažniausiai šis signalas pradedamas vykdyti (vykdomas perėjimas į aukštą signalo frontą) vos tik startavus realizuotai valdymo sistemai. Šiuo atveju po 400 ns „reset“ signalas gražinamas į žemą frontą, tada startuoja dauguma TOP schemos funkcijų.



34 pav. TOP schemos sintezavimo rezultatai

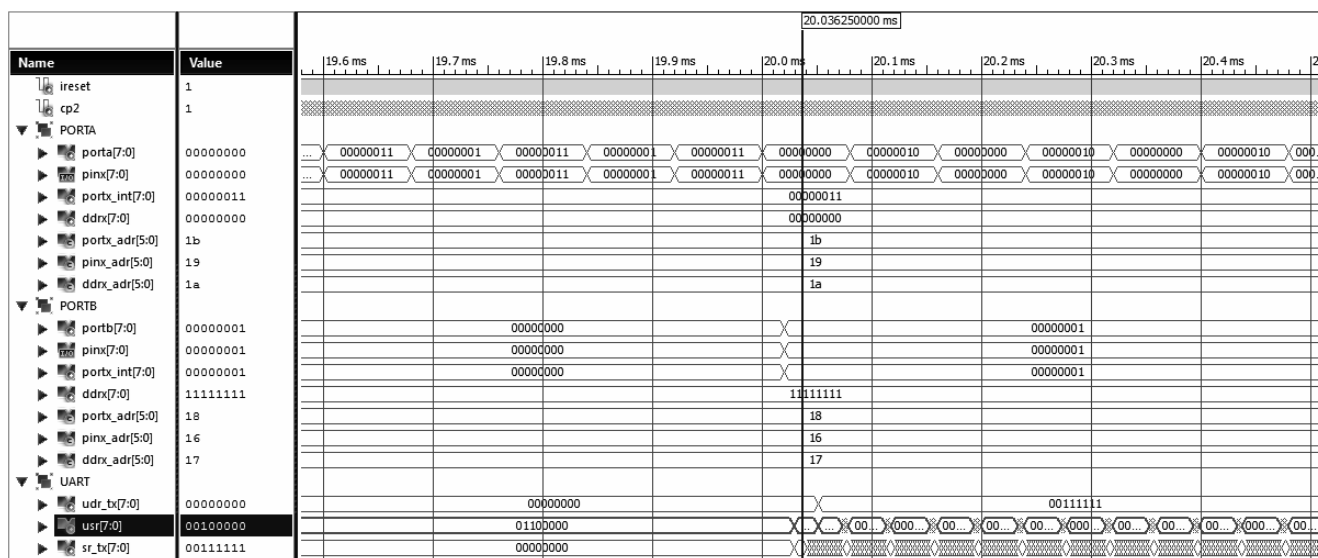
Šaltinis: sudaryta autoriaus

Modeliavimo rezultatuose 33 pav. pateiktas „send_data“ signalas, kuris informuoja UART_TX schemą apie VGMĮ įtaiso atliktą impulsinių signalų skaičiavimo funkciją. Tokiu būdu vartotojui perduodami ne tik VGMĮ, bet ir sekančių schemų rezultatai. Informacinių bitų perdavimas vykdomas iš UART_TX schemos 40 bitų buferio „buf“ per „tx_ser_out“ išeities signalą (pačiau apie UART_TX schemos buferį aprašyta poskyryje „Tiriamąo darbo metodai“), naudojant „count“ skaitiklį. Šio skaitiklio pagalba yra vykdomas 40 žingsnių buferio informacinių bitų postūmis per „tx_ser_out“ išeities signalą. Pagal modeliavimo rezultatus matyti, kad informacinių bitų postūmis, t.y. perdavimas vykdomas esant VGMĮ schemos generuojamam žemo fronto „send_data“ signalui.

Toliau 34 paveikslėlyje pateikti TOP schemos sintezavimo rezultatai. Paveikslėlio kairėje pusėje matyti bendrą sintezuotos schemos bloko vaizdą, kuris sudarytas iš vidinių loginių elementų. Šio bloko kairėje pusėje pavaizduoti anksčiau aptartų schemų (VGMĮ, VKRĮ, BKĮ, GĮMĮ ir UART_TX) išoriniai įėjčių signalai. Atitinkamai, bloko dešinėje pusėje matyti schemų išeičių signalus. Toliau 34 paveikslėlio dešinėje pusėje pavaizduotas sintezuotos pagrindinės TOP schemos išskleistas vaizdas. Taigi, kaip ir paminėta TOP blokas sudarytas iš VGMĮ, VKRĮ, BKĮ, GĮMĮ ir UART_TX funkcinių elementų bei ryšių tarp jų.

2.2.3.2 Vėjo jėgainės valdymo sistemos Nr.: 2 realizacija

VGMĮ algoritmo modeliavimo rezultatai pateikti žemiau esančiuose paveikslėliuose (žiūr. 35 ir 36 pav.). Šiuose paveikslėliuose pavaizduoti pagrindiniai (Atmega103 realizuoto FPGA matricoje mikrovaldiklio) registrai bei signalų kitimas laiko intervale.



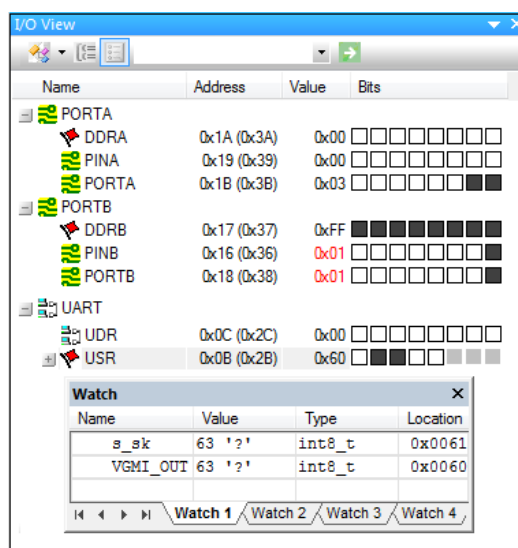
35 pav. VGMĮ algoritmo modeliavimo rezultatai (1)

Šaltinis: sudaryta autoriaus

Taigi matyti „ireset“ „cp2“, „PORTA“, „PORTB“ ir „UART“ registų bei signalų sąrašą. Nagrinėjant pirmuosius du signalus, galima trumpai paminėti, kad „ireset“ skirtas algoritmo perkrovai (paleidimui iš naujo), o „cp2“ – taktinio dažnio (šiuo atveju 4MHz) sinchroninis signalas, kuris naudojamas algoritmo procedūrų vykdymui.

Toliau 35 paveikslėlyje pateikti Atmega103 mikrovaldiklio „PORTA“ ir „PORTB“ registrai, kur VGMĮ algoritmo atveju „PORTA“ registro du bitai naudojami įėjties signalams. Tam tikslui pasiekti, rašant valdymo algoritmo išėjties kodą, būtina deaktyvuoti „DDRA“ registro bitus (nustatant į 0x00) bei „PORTA“ parinkti dvi įėjties, kurios algoritme bus naudojamos („PORTA“ registre aktyvuojami du jauniausi bitai, nustatant į 0x03). Atitinkamai „PORTB“ registro jauniausias bitas naudojamas išėjties signalui (aktyvuojamas „DDRB“ į 0xFF bei parenkamas bitas 0x01). Mikrovaldiklio Atmega103 architektūroje „PORTA“ registro 8 bitai saugomi atminties bloke 0x1b adresu, „PINA“ – 0x19 ir „DDRA“ – 0x1a adresu, o „PORTB“ – 0x18, „PINB“ – 0x16 ir „DDRB“ – 0x17 atminties bloko adresais.

Pagal 35 paveikslėlį matyti laiko intervalo ištrauką, kurioje atsispindi VGMĮ algoritmas vykdymas. Paminėta, kad „PORTA“ du bitai naudojami įeičių signalams, taigi jauniausias bitas naudojamas sinchroniniam signalui, kurio periodas 2 sekundės (pusperiodis 1 sekundė). Šio signalo dėka, aktyvuojamas anemometro generuojamų impulsinių signalų skaičiavimas. Sekantis „PORTA“ bitas skirtas anemometro impulsiniam signalui priimti. Atitinkamai „PORTB“ registro jauniausias bitas yra naudojamas rotoriaus sukimosi momento apribojimui, kuris aktyvuojamas pasiekus kritinį vėjo greitį (25 m/s – t.y 63 impulsai per 1 sekundę). Paveikslėliuose (žiūr. 35 ir 36 pav.) matyti valdymo rezultatus pasiekus paminėtą kritinį vėjo greitį (35 pav. pavaizduoti laiko modeliavimo rezultatai laiko intervalu – ISIM progr. įranga, o 36 pav. – naudojant AVR Studio 4.16 progr. įrangą).

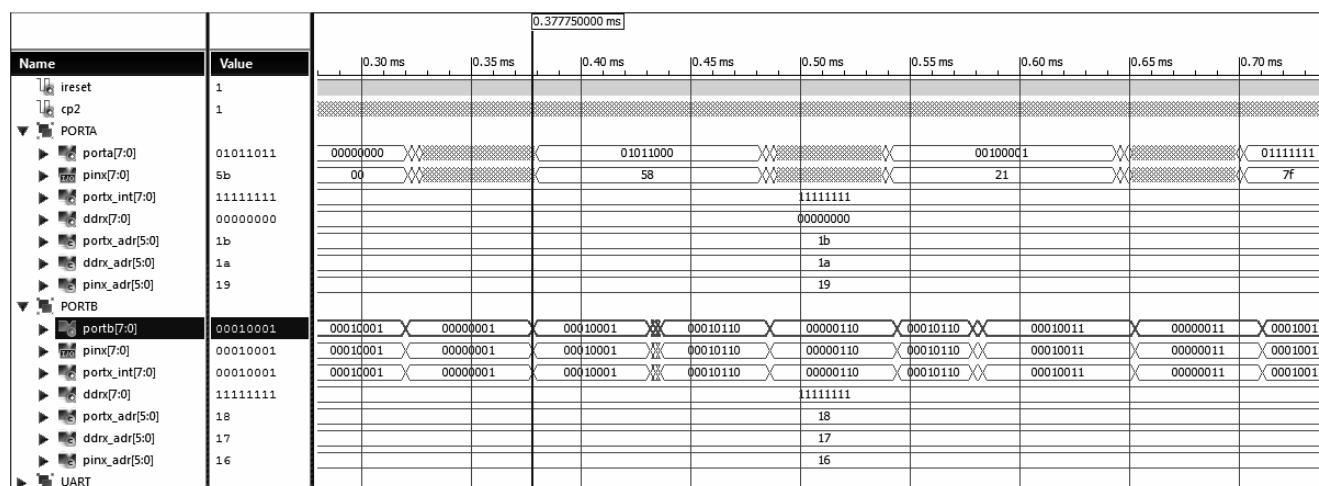


36 pav. VGMĮ algoritmo modeliavimo rezultatai (2)

Šaltinis: sudaryta autoriaus

Modeliavimo rezultatuose (žiūr. 35 pav.) tap pat atsispindi „UART“ registrai, kur į „UDR“ registrą yra patalpinamas VGMĮ algoritmo suskaičiuotas galutinis rezultatas, tokiu būdu per „sr_tx“ informacija yra išstumiami vartotojui (į personalinį kompiuterį, rx schema). Galutinai įvykdžius visų informacinių bitų postūmį, aktyvuojamas „USR“ registro ketvirtas bitas (TXC – UART Transmit Complete), informuojantis apie informacinių bitų įvykdytą perdavimą. Tinkamai perdavus bitus bei gavus sekantį VGMĮ schemos suskaičiuotą rezultatą, išvalomas UART duomenų registras, to pasekoje aktyvuojamas trečias „USR“ registro bitas (UDRE – UART Data Registry Empty) (žiūr. 36 pav.).

VKRĮ algoritmo modeliavimo rezultatai pateikti žemiau esančiuose paveikslėliuose (žiūr. 37 ir 38 pav.). Šiuose paveikslėliuose pavaizduoti pagrindiniai Atmega103 mikrovaldiklio, kurios architektūra įgyvendinta FPGA matricoje, naudojami registrai bei signalai. Nagrinėjant pirmuosius du signalus, galima trumpai paminėti, kad „ireset“ skirtas algoritmo perkrovai (paleidimui iš naujo), o „cp2“ – taktinio dažnio (šiuo atveju 4MHz) sinchroninis signalas, kuris naudojamas algoritmo procedūrų vykdymui.

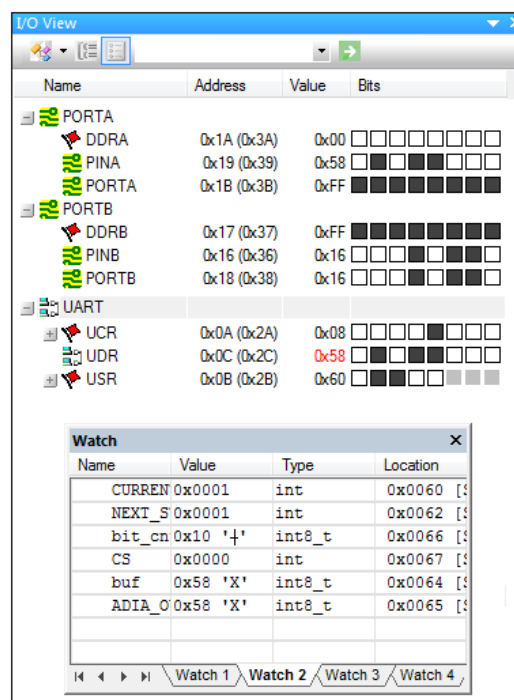


37 pav. VKRĮ algoritmo modeliavimo rezultatai (1)

Šaltinis: sudaryta autoriaus

Toliau 37 paveikslėlyje pateikti Atmega103 mikrovaldiklio „PORTA“ ir „PORTB“ registrai, kur VKRĮ algoritmo atveju „PORTA“ registro 8 bitai naudojami įeities signalams. Tam tikslui pasiekti, rašant valdymo algoritmo išeities kodą, būtina deaktyvuoti „DDRA“ registro bitus (nustatant į 0x00) bei „PORTA“ parinkti 8 įeitys, kurios bus naudojamos algoritme („PORTA“ registre aktyvuojami visi bitai, nustatant į 0xFF). Atitinkamai „PORTB“ registro penki bitai naudojami išeities signalams (aktyvuojamas „DDRB“ į 0xFF). Mikrovaldiklio Atmega103 architektūroje „PORTA“ registro 8 bitai saugomi atminties bloke 0x1b adresu, „PINA“ – 0x19 ir „DDRA“ – 0x1a adresu, o „PORTB“ – 0x18, „PINB“ – 0x16 ir „DDRB“ – 0x17 atminties bloko adresais.

Pagal 37 paveikslėlį matyti laiko intervalo ištrauką, kurioje atsispindi dalis VKRĮ algoritmo vykdymo. Paminėta, kad „PORTA“ 8 bitai naudojami įėjimų signalams, taigi šiuo atveju šie bitai skirti ADC keitiklio analoginio signalo (vėjo krypties 0...360°) konvertuotai skaitmeninei reikšmei priimti. Atitinkamai „PORTB“ registro pirmieji keturi bitai (mikrovaldiklio išėjimo signalai) skirti žingsninio variklio valdymui. Penktasis „PORTB“ bitas naudojamas „cs“ išėjimo signalui, kuris skirtas paleisti / stabdyti ADC keitiklio darbą. Kai „cs“ signalas (penktas „PORTB“ bitas) yra žemo fronto, tada ADC keitiklis vykdo analoginės reikšmės konvertavimą į skaitmeninę, priešingu atveju (esant aukšto fronto „cs“ signalui) ADC keitiklio darbas baigiamas ir konvertuota skaitmeninė informacija perduodama CPU vykdymo įtaisui bei UART registru. Paveikslėlyje (žiūr. 37 pav.) matyti valdymo rezultatus esant vėjo krypties intervalui nuo 135°...157° (šiuo atveju „PINA“ reikšmė yra 0x58), to pasekoje aktyvuojamos žingsninio variklio valdymo „PINB“ išėjimo „0110“. Toliau esant vėjo krypties intervalui nuo 45°...67° („PINA“ reikšmė yra 0x21) – aktyvuojamos žingsninio variklio valdymo „PINB“ išėjimo „0011“ ir esant vėjo krypties intervalui 157°...180° („PINA“ reikšmė yra 0x7F) – nustatomos „0010“ išėjimo. Pirmuoju valdymo atveju horizontaliosios turbinos ašies ir oro srauto krypties kampas pasukamas 135° kampu, sekančiu vėjo turbinos kryptis pasukama 45° kampu ir trečiuoju valdymo atveju kampas pasukamas 180° kampu.



38 pav. VKRĮ algoritmo modeliavimo rezultatai (2)

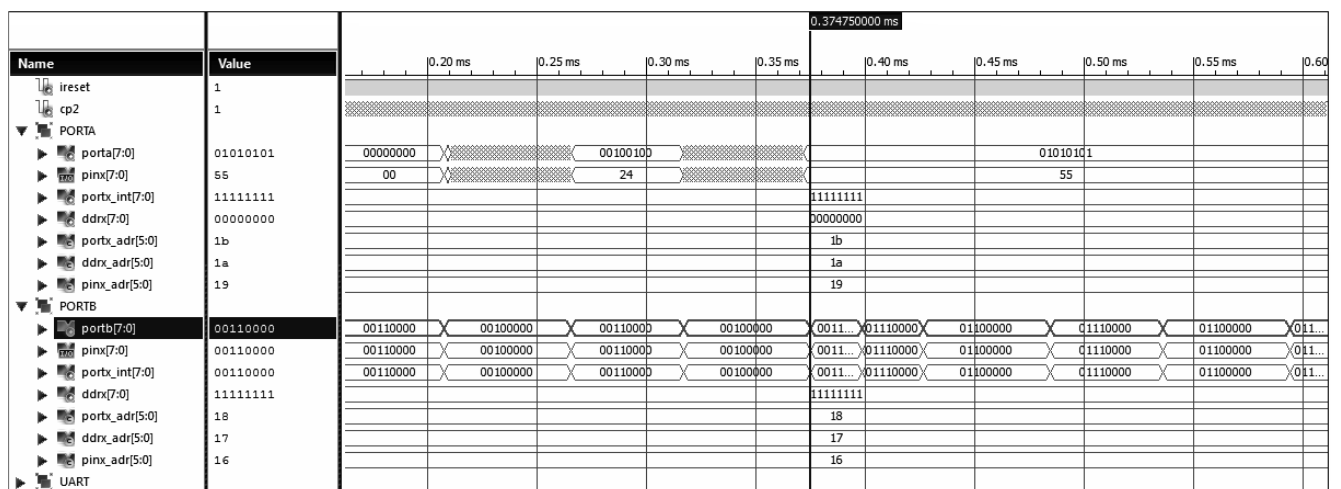
Šaltinis: sudaryta autoriaus

Viršuje esančiame paveikslėlyje (žiūr. 38 pav.) matyti algoritmo vykdymą pagal aprašytą pirmąjį valdymo atvejį (t.y. esant vėjo krypties intervalui nuo 135°...157°). Taip pat šiame modeliavimo

rezultatų atvaizde pavaizduotas UART funkcijos suveikimas. Esantys modeliavimo rezultatai, kurie pateikti laiko intervalo ištraukoje (žiūr. 37 pav.) sumodeliuoti naudojant Xilinx kompanijos ISIM programinę įrangą, o 36 pav. – naudojant Atmel kompanijos AVR Studio 4.16 programinę įrangą.

BKĮ algoritmo modeliavimo rezultatai pateikti žemiau esančiuose paveikslėliuose (žiūr. 39 ir 40 pav.). Šiuose paveikslėliuose pavaizduoti pagrindiniai Atmega103 mikrovaldiklio, kurios architektūra įgyvendinta FPGA matricoje, naudojami registrai bei signalai. Nagrinėjant pirmuosius du signalus, galima trumpai paminėti, kad „ireset“ skirtas algoritmo perkrovai (paleidimui iš naujo), o „cp2“ – taktinio dažnio (šiuo atveju 4MHz) sinchroninis signalas, kuris naudojamas algoritmo procedūrų vykdymui.

Toliau 39 paveikslėlyje pateikti Atmega103 mikrovaldiklio „PORTA“ ir „PORTB“ registrai, kur BKĮ algoritmo atveju „PORTA“ registro 8 bitai naudojami įėjties signalams. Tam tikslui pasiekti, rašant valdymo algoritmo išėjties kodą, būtina deaktyvuoti „DDRA“ registro bitus (nustatant į 0x00) bei „PORTA“ parinkti 8 įėjties, kurios bus naudojamos algoritme („PORTA“ registre aktyvuojami visi bitai, nustatant į 0xFF). Atitinkamai „PORTB“ registro paskutiniai keturi bitai naudojami išėjties signalams (aktyvuojamas „DDRB“ į 0xFF).



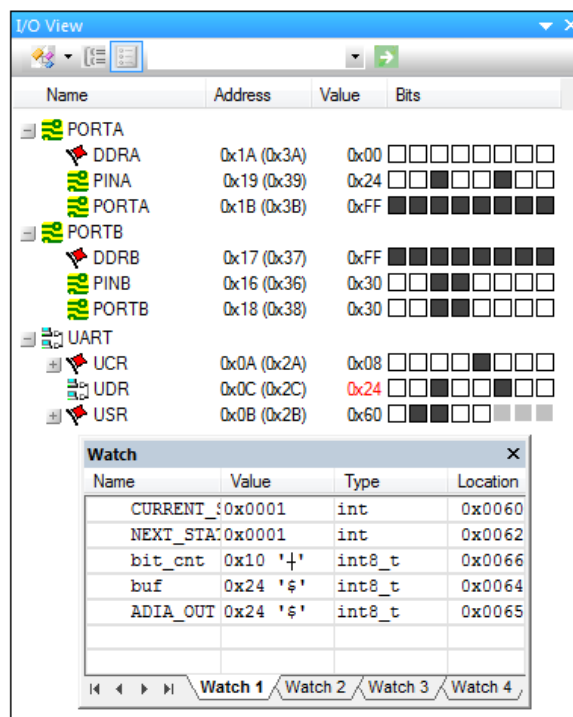
39 pav. BKĮ algoritmo modeliavimo rezultatai (1)

Šaltinis: sudaryta autoriaus

Mikrovaldiklio Atmega103 architektūroje „PORTA“ registro 8 bitai saugomi atminties bloke 0x1b adresu, „PINA“ – 0x19 ir „DDRA“ – 0x1a adresu, o „PORTB“ – 0x18, „PINB“ – 0x16 ir „DDRB“ – 0x17 atminties bloko adresais.

Pagal 39 paveikslėlį matyti laiko intervalo ištrauką, kurioje atsispindi dalis BKĮ algoritmo vykdymo. Paminėta, kad „PORTA“ 8 bitai naudojami įėjčių signalams, taigi šiuo atveju šie bitai skirti ADC keitiklio analoginio signalo (baterijų bloko el. talpos 0...100%) konvertuotai skaitmeninei reikšmei priimti. Atitinkamai „PORTB“ registro paskutiniai trys bitai (mikrovaldiklio išėjimo

signalai) skirti baterijų bloko kovos leidimų aktyvavimui. Penktasis „PORTB“ bitas naudojamas „cs“ išeities signalui, kuris skirtas paleisti / stabdyti ADC keitiklio darbą. Kai „cs“ signalas (penktas „PORTB“ bitas) yra žemo fronto, tada ADC keitiklis vykdo analoginės reikšmės konvertavimą į skaitmeninę, priešingu atveju (esant aukšto fronto „cs“ signalui) ADC keitiklio darbas baigiamas ir konvertuota skaitmeninė informacija perduodama CPU vykdymo įtaisu bei UART registrui. Paveikslėlyje (žiūr. 39 pav.) matyti valdymo rezultatus esant baterijų blokui įkrautam iki 20% (šiuo atveju „PINA“ reikšmė yra 0x24), to pasekoje aktyvuojamos baterijų bloko krovos leidimų „PINB“ išeitys „001“. Toliau esant baterijų blokui įkrautam iki 40% („PINA“ reikšmė yra 0x55) – aktyvuojamos žingsninio variklio valdymo „PINB“ išeitys „011. Pirmuoju valdymo atveju yra formuojamas leidimas vykdyti baterijų blokų krovą, uždraudžiama vartoti elektros energiją iš baterijų, bei eikvoti perteklinę generuojamą el. energiją. Sekančiu atveju yra formuojami leidimai vykdyti baterijų blokų krovą bei vartoti elektros energiją iš baterijų, uždraudžiama eikvoti perteklinę generuojamą el. energiją vėjo.



40 pav. BKĮ algoritmo modeliavimo rezultatai (2)

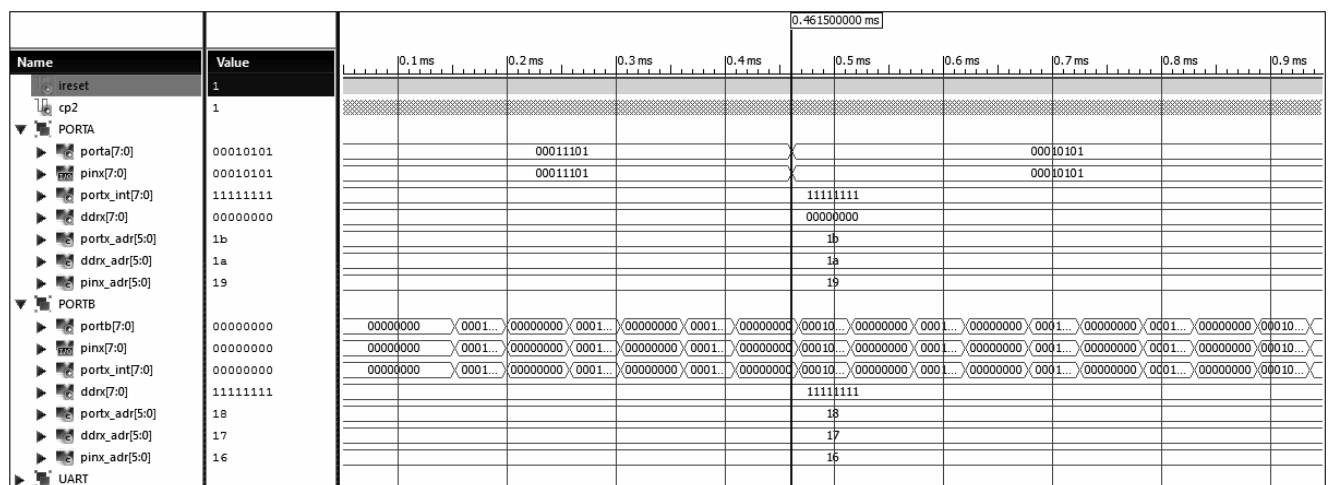
Šaltinis: sudaryta autoriaus

Viršuje esančiame paveikslėlyje (žiūr. 40 pav.) matyti algoritmo vykdymą pagal aprašytą pirmąjį valdymo atvejį (t.y. esant baterijų blokui įkrautam iki 20%). Taip pat šiame modeliavimo rezultatu atvaizde pavaizduotas UART funkcijos suveikimas. Esantys modeliavimo rezultatai, kurie pateikti laiko intervalo ištraukoje (žiūr. 39 pav.) sumodeliuoti naudojant Xilinx kompanijos ISIM programinę įrangą, o 40 pav. – naudojant Atmel kompanijos AVR Studio 4.16 programinę įrangą.

GĮMĮ algoritmo modeliavimo rezultatai pateikti žemiau esančiuose paveikslėliuose (žiūr. 41 ir 42 pav.). Šiuose paveikslėliuose pavaizduoti pagrindiniai Atmega103 mikrovaldiklio, kurios architektūra įgyvendinta FPGA matricoje, naudojami registrai bei signalai. Nagrinėjant pirmuosius du signalus, galima trumpai paminėti, kad „ireset“ skirtas algoritmo perkrovai (paleidimui iš naujo), o „cp2“ – taktinio dažnio (šiuo atveju 4MHz) sinchroninis signalas, kuris naudojamas algoritmo procedūrų vykdymui.

Toliau 41 paveikslėlyje pateikti Atmega103 mikrovaldiklio „PORTA“ ir „PORTB“ registrai, kur GĮMĮ algoritmo atveju „PORTA“ registro 8 bitai naudojami įėjties signalams. Tam tikslui pasiekti, rašant valdymo algoritmo išeities kodą, būtina deaktyvuoti „DDRA“ registro bitus (nustatant į 0x00) bei „PORTA“ parinkti 8 įeitys, kurios bus naudojamos algoritme („PORTA“ registre aktyvuojami visi bitai, nustatant į 0xFF). Atitinkamai „PORTB“ registro penktas bitas naudojamas išeities signalui (aktyvuojamas „DDRB“ į 0xFF).

Mikrovaldiklio Atmega103 architektūroje „PORTA“ registro 8 bitai saugomi atminties bloke 0x1b adresu, „PINA“ – 0x19 ir „DDRA“ – 0x1a adresu, o „PORTB“ – 0x18, „PINB“ – 0x16 ir „DDRB“ – 0x17 atminties bloko adresais.

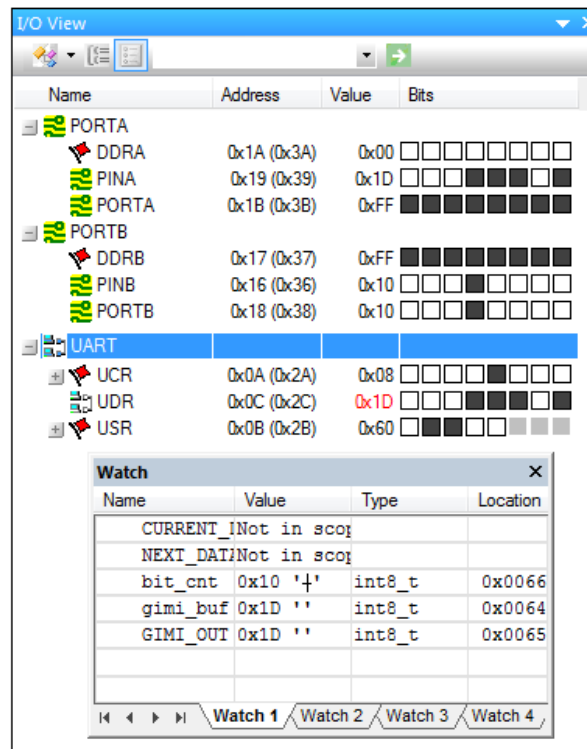


41 pav. GĮMĮ algoritmo modeliavimo rezultatai (1)

Šaltinis: sudaryta autoriaus

Pagal 41 paveikslėlį matyti laiko intervalo ištrauką, kurioje atsispindi dalis GĮMĮ algoritmo vykdymo. Paminėta, kad „PORTA“ 8 bitai naudojami įėjčių signalams, taigi šiuo atveju šie bitai skirti ADC keitiklio analoginio signalo (generuojamos įtampos) konvertuotai skaitmeninei reikšmei priimti. Atitinkamai „PORTB“ penktasis bitas naudojamas „cs“ išeities signalui, kuris skirtas paleisti / stabdyti ADC keitiklio darbą. Kai „cs“ signalas (penktas „PORTB“ bitas) yra žemo fronto, tada ADC keitiklis vykdo analoginės reikšmės konvertavimą į skaitmeninę, priešingu atveju (esant aukšto fronto „cs“

signalui) ADC keitiklio darbas baigiamas ir konvertuota skaitmeninė informacija perduodama CPU vykdymo įtaisu bei UART registrui.

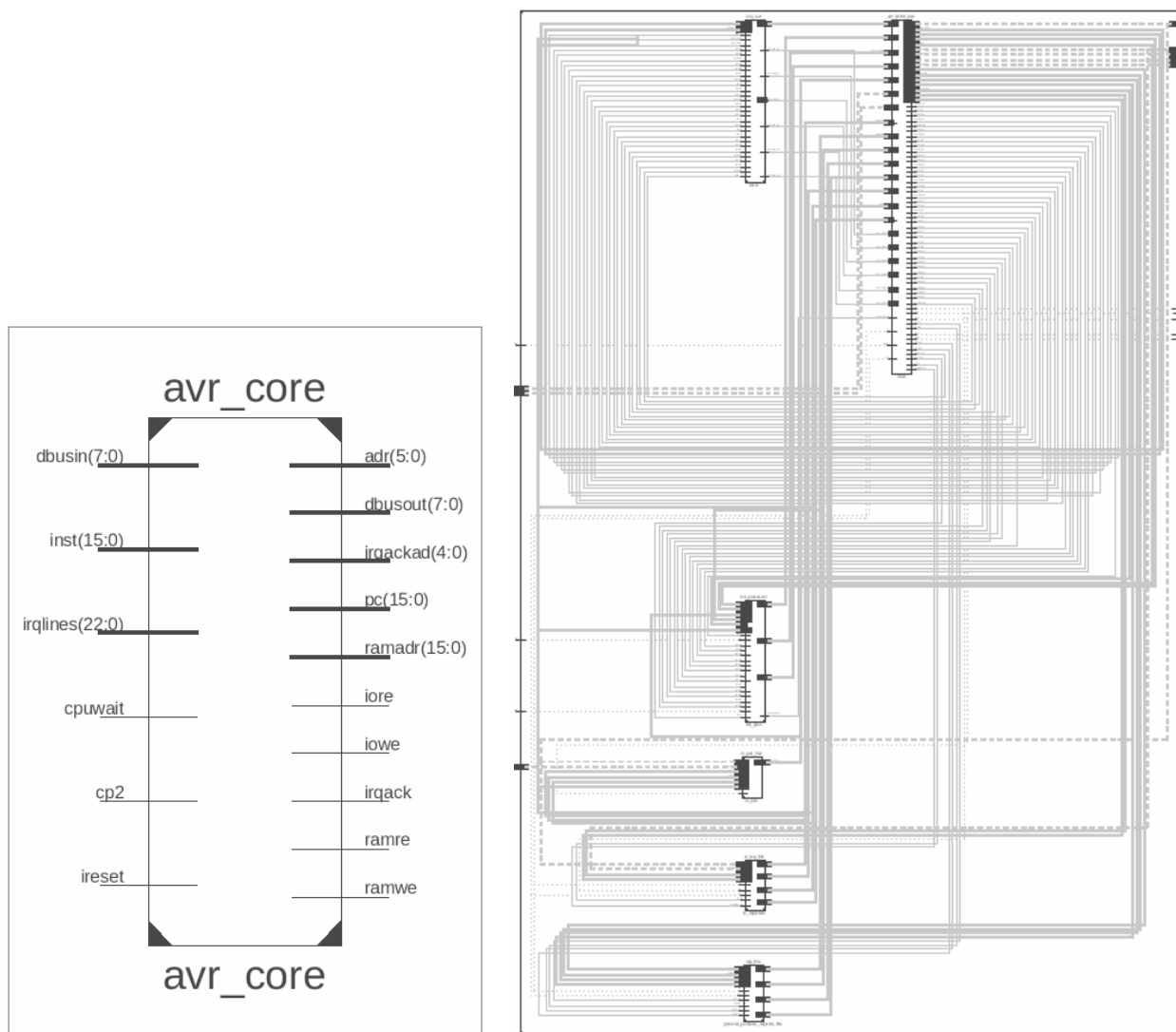


42 pav. GĪMĪ algoritmo modeliavimo rezultatai (2)
Šaltinis: sudaryta autoriaus

Viršuje esančiame paveikslėlyje (žiūr. 42 pav.) pavaizduotas UART funkcijos suveikimas, kur į „UDR“ registrą yra patalpinamas GĪMĪ algoritmo suskaičiuotas galutinis rezultatas, tokiu būdu vartotojui yra išstumiamas informacija (į personalinį kompiuterį, rx schema). Galutinai įvykdžius visų informacinių bitų postūmį, aktyvuojamas „USR“ registro ketvirtas bitas (TXC – UART Transmit Complete), informuojantis apie informacinių bitų įvykdytą perdavimą. Tinkamai perdavus bitus bei gavus sekantį VGMĪ schemos suskaičiuotą rezultatą, išvalomas UART duomenų registras, to pasekoje aktyvuojamas trečias „USR“ registro bitas (UDRE – UART Data Registry Empty) (žiūr. 42 pav.).

Esantys modeliavimo rezultatai, kurie pateikti laiko intervalo atkarpoje (žiūr. 41 pav.) sumodeliuoti naudojant Xilinx kompanijos ISIM programinę įrangą, o 42 pav. – naudojant Atmel kompanijos AVR Studio 4.16 programinę įrangą.

Mikrovaldiklio Atmega103 architektūros sintezavimo rezultatai pateikti 43 paveikslėlyje. Šio paveikslėlio kairėje pusėje matyti bendrą, supaprastintą AVR Atmega103 mikrovaldiklio vaizdą, kur kairėje pusėje atvaizduojami įeities signalai, o dešinėje pusėje atitinkamai – išeities signalai. Paveikslėlio dešinėje pusėje pateiktas sintezuoto Atmega103 mikrovaldiklio bloko išsklęstas vaizdas, taigi matyti, kad sistemos funkcionalumą sudaro eilę elementų bei ryšių tarp jų.



43 pav. Atmega103 mikrovaldiklio architektūros sintezavimo rezultatai

Šaltinis: sudaryta autoriaus

Aukščiau esančiame paveikslėlyje (žiūr. 43 pav.), esančioje sintezuotoje Atmega103 mikrovaldiklio architektūroje, naudojami pagrindiniai elementai, kurie reikalingi šiame darbe aptartų vėjo jėgainės valdymo sistemų realizacijai atlikti. Taigi šie elementai susideda iš: centrinio vykdymo įtaiso CPU, atminties bloką (programinės atminties, sparčiosios, laikinosios informacijos atminties elementu), duomenų magistralės, programuojamų I/O įeičių/išeičių, laiko skaitiklio, UART elemento ir pan.

2.2.3.3 Realizuotų vėjo jėgainės valdymo sistemų rezultatų palyginimas

Atlikę sistemos realizaciją ir gavę rezultatus, galima palyginti skirtingas vėjo jėgainės valdymo sistemas. Taigi tyrimas atliktas dviejų tipų sistemoms: vėjo jėgainės valdymo sistema Nr.: 1 ir vėjo jėgainės valdymo sistema Nr.: 2, kur pirmojoje sistemoje valdymo algoritmas realizuotas tiesiogiai

FPGA matricoje, o antrosios valdymo sistemos atveju – valdymo algoritmas patalpintas FPGA matricoje įgyvendinto mikrovaldiklio Atmega103 programinėje atmintyje. Tokiu būdu tyrimo metu atliktas dviejų valdymo sistemų architektūrų palyginimas. Siekiant atlikti tikslingą šių sistemų tarpusavio palyginimą, naudojamas vienodas valdymo algoritmas, kuris vykdomas tokiu pačiu taktiniu dažniu (t.y. 4MHz) .

Vėjo jėgainių valdymo sistemų tarpusavio palyginimas pradedamas nuo valdymo algoritmo VGMĮ dalies. Taigi žemiau esančioje lentelėje (žiūr. 3 lent.) matyti funkcijas bei jų įvykdymo laikus skirtingose realizuotose valdymo sistemose. Nagrinėjant 3 lentelės pirmąją VGMĮ valdymo algoritmo komandą „ Impulsinių signalų skaičiavimo trukmė“, matyti, kad šios komandos vykdymo laikas yra vienodas (t.y. viena sekundė). Taip yra todėl, kad anemometro generuojami impulsiniai signalai turi būti išreikšti dažniu [Hz], o dažnis (išreikštas hercais) – tai fizikinis dydis, rodantis, kiek kartų įvykis pasikartoja per laiko vienetą (šiuo atveju sekundę), todėl šios komandos vykdymo laikas (nepriklausomai nuo sistemos tipo ar glimybės) visada yra pastovus, trunkantis lygiai vieną sekundę.

VYKDOMOS KOMANDOS	FPGA ¹ VGMĮ	MCU ² VGMĮ
Impulsinių signalų skaičiavimo trukmė	1000000us	1000000us
Rezultato įrašymas į buferį	~ 0 us	8 ÷ 19.25 us
Vykdoma funkcija, pagal buferyje patalpintą rezultatą	~ 0 us	19.25 ÷ 23.25 us
Rezultato perdavimas iš buferio į UART	0.1 us	12.25 us
Perėjimas į sekančią schemą	-	11.75 us

3 lent. VGMĮ modeliavimo rezultatų palyginimas

Šaltinis: sudaryta autoriaus

„Rezultato įrašymas į buferį“ komandos vykdymo metu, suskaičiuota anemometro generuojamo signalo dažnio reikšmė yra patalpinama buferyje. Toliau atliekamas sistemos valdymas pagal buferyje patalpintą informaciją (ši komanda lentelėje pavadinta „Vykdoma funkcija, pagal buferyje patalpintą rezultatą“). Taigi šios dvi funkcijos (arba komandos) abiejuose valdymo sistemose atliekamos skirtingais laiko intervalais. Matyti, kad įvykdymo trukmė apie dešimt kartų ilgesnė MCU (Micro – controller Unit) Atmega103 mikrovaldiklyje nei FPGA matricoje. Taip yra todėl, kad mikrovaldiklyje visos procedūros bei funkcijos yra vykdomos nuosekliai, o FPGA matricoje procedūros atliekamos lygiagrečiai (nuosekliai vykdomos tik procedūroje esančios funkcijos ar komandos). Pagal 3 lentelėje esančius rezultatus galima pastebėti, kad ši savybė algoritmo vykdymo atžvilgiu yra daug pranašesnė FPGA matricoje.

„Rezultato perdavimas iš buferio į UART“ komandos metu perduodami informaciniai bitai iš vidinio VGMĮ buferio į UART funkcinį elementą. Šios komandos vykdymui, mikrovaldiklio

¹ Vėjo jėgainės realizuota valdymo sistema Nr.: 1

² Vėjo jėgainės realizuota valdymo sistema Nr.: 2

architektūroje informacijos perdavimo trukmė priklauso ne tik nuo nuoseklaus procedūrų vykdymo, tačiau ir nuo UART funkcijos vykdymo dažnio „baudclk“. FPGA matricos atveju vėlavimas priklauso tik nuo „baudclk“ signalo taktinio dažnio. Toliau mikrovaldiklio Atmega103 atveju, įvykdžius visas VGMĮ funkcijas, atliekamas perėjimas į sekančią valdymo algoritmo dalį. Kaip matyti 3 lentelėje šis perėjimas trunka apie 11.75 us. Valdymo algoritmo realizuoto FPGA matricoje perėjimas nėra vykdomas, nes visos valdymo algoritmo dalys (procedūros) vykdomos lygiagrečiai.

Sekancioje lentelėse (žiūr. 4 lent.) pateiktos vėjo jėgainės valdymo algoritmo (VKRĮ ir BKĮ dalių) pagrindinės vykdomos komandos. VKRĮ ir BKĮ valdymo algoritmo dalyje pirmiausia atliekamas analoginio signalo konvertavimas į skaitmeninio tipo signalą (ADC keitiklio pagalba). Šiuo atveju ADC keitiklis yra valdomas „CS“ signalo pagalba, t.y. esant „CS“ žemo fronto signalui, ADC keitiklis atlieka analoginio signalo konversiją, taigi šiuo momentu vykdomas galutinio rezultato – skaitmeninio signalo atrinkimas. Pagal pateiktą lentelę matyti, kad mikrovaldiklio Atmega103 atveju ši komanda yra vykdoma žymiai ilgiau nei FPGA matricoje.

VYKDOMOS KOMANDOS	FPGA VKRĮ	MCU VKRĮ
Signalų atrinkimo trukmė (esant CS = '0')	4 us	58 us
Rezultato įrašymas į buferį	0.75 us	3 us
Vykdoma funkcija, pagal buferyje patalpintą rezultatą	0.25us	23.25 ÷ 113.25 us
Rezultato perdavimas iš buferio į UART	0.87us	72.25 us
Perėjimas į sekančią schemą	-	12.25 us

VYKDOMOS KOMANDOS	FPGA BKĮ	MCU BKĮ
Signalų atrinkimo trukmė (esant CS = '0')	4 us	58 us
Rezultato įrašymas į buferį	0.75 us	3 us
Vykdoma funkcija, pagal buferyje patalpintą rezultatą	0.25 us	18.75 ÷ 27.75 us
Rezultato perdavimas iš buferio į UART	0.87us	32.25 us
Perėjimas į sekančią schemą	-	8.5 us

4 lent. VKRĮ ir BKĮ modeliavimo rezultatų palyginimai

Šaltinis: sudaryta autoriaus

„Rezultato įrašymas į buferį“ komandos vykdymo metu (žiūr. 4 lent.), konvertuota skaitmeninio signalo – informacinių bitų reikšmė patalpinama buferyje. Po to atliekamas sistemos valdymas pagal buferyje patalpintą informaciją (ši komanda lentelėje pavadinta „Vykdoma funkcija, pagal buferyje patalpintą rezultatą“). Taigi šios dvi funkcijos (arba komandos) abiejuose valdymo sistemose atliekamos skirtingais laiko intervalais. Matyti, kad įvykdymo trukmė yra žymiai ilgesnė MCU (Micro – Controller Unit) Atmega103 mikrovaldiklyje nei FPGA matricoje todėl, kad mikrovaldiklyje visos procedūros bei funkcijos yra vykdomos nuosekliai, o FPGA matricoje procedūros atliekamos

lygiagrečiai (nuosekliai vykdomos tik procedūroje esančios funkcijos ar komandos). Kaip ir prieš tai nagrinėtu atveju galima teigti, kad ši savybė algoritmo vykdymo atžvilgiu yra daug pranašesnė FPGA matricoje.

Analogiškai aptartai valdymo algoritmo VGMĮ dalies atvejui, „Rezultato perdavimas iš buferio į UART“ komandos metu perduodama informaciniai bitai iš vidinio VKRĮ (arba BKĮ) buferio į UART funkcinį elementą. Šios komandos vykdymui, mikrovaldiklio architektūroje, informacijos perdavimo trukmė priklauso ne tik nuo nuoseklaus procedūrų vykdymo, tačiau ir nuo UART funkcijos vykdymo dažnio „baudclk“. FPGA matricos atveju vėlavimas priklauso tik nuo „baudclk“ signalo taktinio dažnio. Toliau mikrovaldiklio Atmega103 atveju, įvykdžius visas VKRĮ funkcijas, atliekamas perėjimas į sekančią valdymo algoritmo dalį – BKĮ. Valdymo algoritmo realizuoto FPGA matricoje perėjimas nėra vykdomas, nes visos valdymo algoritmo dalys vykdomos lygiagrečiai.

Valdymo algoritmo GĮMĮ dalies atveju lentelėje esančios komandos „Signalų atrinkimo trukmė“ funkcija yra identiška aptartai VKRĮ ir BKĮ dalių funkcijai. Taigi ADC keitiklis atlieka analoginio signalo (generuojamos įtampos reikšmės) konversiją, šios komandos vykdymo momentu atliekamas galutinio rezultato – skaitmeninio signalo atrinkimas. Pagal pateiktą lentelę matyti, kad mikrovaldiklio Atmega103 atveju ši komanda yra vykdoma žymiai ilgiau nei FPGA matricoje.

VYKDOMOS KOMANDOS	FPGA GĮMĮ	MCU GĮMĮ
Signalų atrinkimo trukmė (esant CS = '0')	4 us	58 us
Informacijos įrašymas į buferį	0.75 us	2.5 us
Informacijos perdavimas iš buferio į UART	0.87 us	11.75 us
Perėjimas į sekančią schemą	-	11.75 us

5 lent. GĮMĮ modeliavimo rezultatų palyginimas

Šaltinis: sudaryta autoriaus

„Informacijos įrašymas į buferį“ komandos vykdymo metu (žiūr. 4 lent.), konvertuota skaitmeninio signalo – informacinių bitų reikšmė patalpinama buferyje. Taigi ši funkcija (arba komanda) abiejuose valdymo sistemose atliekama skirtingais laiko intervalais. Matyti, kad įvykdymo trukmė yra žymiai ilgesnė MCU (Micro – controller Unit) Atmega103 mikrovaldiklyje nei FPGA matricoje. Priežastis yra analogiška aptartoms valdymo algoritmo dalims – mikrovaldiklyje visos procedūros bei funkcijos yra vykdomos nuosekliai, o FPGA matricoje procedūros atliekamos lygiagrečiai (nuosekliai vykdomos tik procedūroje esančios funkcijos ar komandos). Kaip ir prieš tai nagrinėtu atveju ši savybė algoritmo vykdymo atžvilgiu yra daug pranašesnė FPGA matricoje.

Žemiau esančioje lentelėje (žiūr. 6 lent.) pateiktas bendras viso vėjo jėgainės valdymo algoritmo (kuris susideda iš dalių VGMĮ, VKRĮ, BKĮ, GĮMĮ bei UART_TX) vykdymo laikas skirtinguose valdymo sistemose.

Matyti, kad šis FPGA matricoje realizuotas valdymo algoritmas yra atliekamas iki 526.75 us greičiau nei MCU mikrovaldiklio architektūroje.

VYKDOMOS KOMANDOS	FPGA TOP	MCU TOP
Impulsinių signalų skaičiavimo trukmė	1000000us	1000000us
Signalų atrinkimo trukmė (esant CS = '0')	12 us	174 us
Rezultatoįrašymas į buferį	0.75 us	16.50÷ 27.75 us
Vykdoma funkcija, pagal buferyje patalpintą rezultatą	0.25 us	61.25÷ 164.25us
Rezultato perdavimas iš buferio į UART	0.87 us	128.50us
Perėjimas į sekančią schemą	-	44.25 us
VISO (be VGMĮ impulsinių signalų skaičiavimo):	12 us	424.50÷ 538.75 us
VISO (su VGMĮ impulsinių signalų skaičiavimu):	1000012.00 us	100424.50 ÷ ÷ 100538.75 us
Skirtumas:	412.50÷ 526.75 us	

6 lent. TOP modeliavimo rezultatų palyginimas

Šaltinis: sudaryta autoriaus

Toliau 7 lentelėje pateikti tiriamų skirtingų valdymo sistemų sintezavimo rezultatai. Matyti, kad valdymo algoritmui vykdyti, MCU mikrovaldiklio architektūra naudoja daug daugiau elementų lyginant su sistema, kurioje tas pats valdymo algoritmas realizuotas tiesiogiai FPGA matricoje.

FPGA TOP		MCU TOP	
<i>NAUDOJAMŲ LOGINIŲ ELEMENTŲ SKAIČIUS</i>			
Loginės dalelės („Slices“)	105 vnt.	Loginės dalelės („Slices“)	402 vnt.
Log. funkcijų lent. („LUTs“)	160 vnt.	Log. funkcijų lent. („LUTs“)	1184 vnt.
D tipo („Flip Flop“) trigeriai	176 vnt.	D tipo („Flip Flop“) trigeriai	484 vnt.
Paskirstytuvai („Mux“)	124 vnt.	Paskirstytuvai („Mux“)	310 vnt.
Skaitikliai („Counters“)	5 vnt.	Skaitikliai („Counters“)	-
Papildomi loginiai ventiliai aritmet. operacijoms atlikti	5 vnt.	Papildomi loginiai ventiliai aritmet. operacijoms atlikti	49 vnt.
IOB įėjimų / išėjimų sk.	65 vnt.	IOB įėjimų / išėjimų sk.	106 vnt.
<i>SIGNALO SKLIDIMO VĒLINIMAS</i>			
Login. element.:	1.335 ns (33.8 %)	Login. element.:	3.955 ns (26.4 %)
Trasoje:	2.613ns (66.2 %)	Trasoje:	10.998 ns (73.6 %)
Viso:	3.948ns (100 %)	Viso:	14.953 ns (100 %)
<i>ENERGIJA</i>			
Statinė energija:	11.14 mW	Statinė energija:	11.14 mW
Dinaminė energija:	0.07 mW	Dinaminė energija:	0.06 mW

7 lent. TOP sintezavimo rezultatų palyginimas

Šaltinis: sudaryta autoriaus

Taip pat 7 lentelėje matome, kad signalų sklaidimo vėlinimas MCU architektūros elementuose ir trasose yra žymiai ilgesnis nei FPGA matricos sintezuotose valdymo algoritmo loginiuose elementuose. Šioje lentelėje pateiktas valdymo sistemų statinės ir dinaminės energijos suvartojimas. Kadangi tyrimas atliktas naudojant tą pačią FPGA matricą ir vienodą taktinį dažnį (4MHz), todėl gauti el. energijos suvartojimo duomenys yra labai panašūs.

Žemiau esančioje 8 lentelėje pateikti realizuotų sistemų bendri duomenys:

<i>REALIZUOTŲ VALDYMO SISTEMŲ BENDRI DUOMENYS</i>	
Įranga:	XILINX
Šeima:	SPARTAN6
Tipo Nr.:	XA6SLX4

8 lent. Vėjo jėgainės realizuotų valdymo sistemų bendri duomenys

Šaltinis: sudaryta autoriaus

Šioje lentelėje pateiktas FPGA matricos tikslus tipas, kuris buvo naudojamas realizuojant vėjo jėgainės valdymo sistemas. Šioje FPGA matricoje maksimalus pagrindinių loginių elementų skaičius yra gana didelis, todėl realizuotos vėjo jėgainių valdymo sistemos išnaudojo ne daugiau nei pusę Spartan6 loginių elementų, pavyzdžiui šioje FPGA matricoje maksimaliai gali būti išnaudotos 600 vnt. loginės dalelės (angl. „Slices“), 4800 vnt. registru, 2400 generuojamų loginių funkcinių lentelių (angl. „LUTs“).

IŠVADOS

Apibendrinant darbe analizuojamas valdymo sistemas bei palyginus gautus rezultatus galima teigti, kad:

1. FPGA matricoje procedūros vykdomos lygiagrečiai, o MCU mikrovaldiklis (RISC architektūra) procesus atlieka nuosekliai, todėl tiesiogiai FPGA matricoje realizuoto valdymo algoritmo (vėjo jėgainės valdymo sistema Nr.: 1) funkcijos įvykdomos daug sparčiau nei MCU mikrovaldiklio architektūros pagrindu realizuotoje sistemoje (vėjo jėgainės valdymo sistema Nr.: 2). Taigi pagal gautus darbo rezultatus matyti, kad vėjo jėgainės valdymo sistemos Nr.: 1 funkcijų (komandų) atlikimo laikas iki 10 ir daugiau kartų yra mažesnis, lyginant su vėjo jėgainės valdymo sistema Nr.: 2;
2. Pagal gautus sistemų sintezavimo rezultatus matyti, kad siekiant realizuoti vienodą valdymo algoritmą, MCU mikrovaldiklio atveju, naudojama daug daugiau loginių elementų. Šiuo atveju visi MCU sintezuoti elementai naudojami RISC architektūros pagrindui sudaryti, o valdymo algoritmas įrašomas į MCU atminties bloką. Taip pat sintezės rezultatuose matyti, kad signalo perdavimo laikas loginiuose elementuose bei ryšiuose tarp jų, yra daug didesnis nei tiesiogiai FPGA matricoje realizuotoje vėjo jėgainės valdymo sistemoje. Šis aspektas taip pat turi didelę įtaką lėtesniam vėjo elektrinės valdymo sistemos Nr.: 2 darbui;
3. FPGA matricoje realizuota vėjo jėgainių valdymo sistema yra daug lankstesnė nei MCU mikrovaldiklio pagrindu įgyvendinta sistema, kadangi FPGA matricos atveju, daugumą valdymo sistemos komponentų galima keisti pagal vartotojo poreikius aprašius juos HDL kalba bei susintezavus FPGA matricoje. Atitinkamai realizuojant bet kokią valdymo algoritmą MCU mikrovaldiklyje, naudojami tie patys RISC architektūros komponentai, tačiau išnaudojamas skirtingas komponentų resursų kiekis (priklausomai nuo sudaryto valdymo algoritmo);
4. Vėjo jėgainių valdymo sistemoms dažnai remiasi multi-procesorinėmis technologijomis, t.y. šios valdymo sistemos susideda iš kelių mikrovaldiklių, kurie atlieka skirtingas funkcijas. Tokiu būdu valdymo sistemose galima vykdyti procedūras lygiagrečiai (kelių mikrovaldiklių pagalba). Kadangi MCU multi-procesorinėse sistemose naudojama daugiau vykdymo įtaisų CPU, komponentų, todėl galima daryti prielaidą, kad atsiranda žymiai didesnė galimų sisteminių gedimų tikimybė. Naudojant FPGA matricų technologijas, galimų sisteminių gedimų skaičių galima ženkliai sumažinti, nes siekiant vėjo jėgainių valdymo algoritmui realizuoti, įmanoma naudoti vieną FPGA lustą, kuris gebėtų atlikti kelių mikrovaldiklių funkcijas (priklausomai nuo FPGA tipo) bei šias funkcijas vykdyti lygiagrečiai.

Atsižvelgus į realizuotų valdymo sistemų gautus rezultatus ir išanalizavus vėjo jėgainių valdymo sistemų tendencijas bei technologijas galima nustatyti optimalų vėjo jėgainių valdymo sistemos variantą.

Nagrinėjant **namų ūkio vėjo jėgainės**, jų technologijas ir valdymo algoritmus, galima teigti, kad tokio tipo vėjo jėgainių optimalus valdymo sistemos variantas – žemo dažnio MCU mikrovaldiklio pagrindu realizuota valdymo sistema. Namų ūkio vėjo elektrinėms nėra būtina naudoti didelę greitaveiką turinčias, FPGA matricos pagrindu realizuotas valdymo sistemas, todėl kad tokio tipo jėgainėms aukšto dažnio valdymas yra nereikalingas, valdymo algoritmo procedūros gali būti atliekamos nuosekliai bei valdymo sistema nėra sudėtinga, t.y. valdymas realizuojamas naudojant vieną MCU RISC architektūros mikrovaldiklį.

Komercinio tipo vėjo jėgainių valdymo sistemos dažnai naudoja multi – procesorines technologijas bei sudėtingus valdymo algoritmus (naudojant PID reguliatorius), siekiant tinkamai kontroliuoti vėjo jėgainės darbą. Dažnai šioms sistemoms yra aktualus greitas sistemos valdymas bei yra pageidautina, kad skirtingi komercinės vėjo jėgainės procesai būtų kontroliuojami lygiagrečiai. Pagal tiriamojo darbo gautus realizuotų sistemų rezultatus, galima teigti, kad tokio tipo vėjo jėgainėms tiktų FPGA matricos pagrindu realizuota valdymo sistema. Naudojant šią sistemą būtų galima procesus vykdyti lygiagrečiai, kurti ir lanksčiai pritaikyti sudėtingus valdymo algoritmus bei sumažinti galimų sisteminių gedimų skaičių.

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Vėjo jėgainė (elektrinė) – įrenginys, kuris verčia kinetinę vėjo energiją į elektros energiją.

Vėjo jėgainės valdymo sistema – įtaisas, kuris skirtas efektyviai (automatiniu būdu) valdyti vėjo jėgainės darbą pagal išorinių prietaisų (ar aplinkos sąlygų) kitimo būsenas, identifikuoti sistemos gedimus ir momentines reikšmes.

PLC (Programmable Logic Computer) valdiklis – skaitmeninis modulinės konstrukcijos programuojamas įtaisas (arba kompiuteris), skirtas įvairaus sudėtingumo pramoninių procesų automatizuotam valdymui.

FPGA (Field Programmable Gate Array) matrica – programuojamos logikos lustas, kuriame HDL sintezuotas kodas realizuojamas loginių ventilių lygmeniu.

VHDL (VHSIC Hardware Description Language) – skaitmeninės elektroninės aparatūros funkcionavimo aprašymo kalba, kuri labai panaši į programavimo kalbą, tačiau šalia konstantų ir kintamųjų turi signalus, su kuriais yra susiejami laiko parametrai ir galima nurodyti operatoriaus vėlinimus.

MCU (Micro – Controller Unit) – tai mažų gabaritų programuojamas įtaisas, kurio sudedamieji elementai panašūs į kompiuterio, t.y. mikrovaldiklis turi procesorių (CPU), atminties elementus (RAM, ROM, EEPROM ir pan.), įėjties/išeities jungtis ir t.t. Šis programuojamas įtaisas naudojamas įterptinėse sistemose (tam tikroms funkcijoms atlikti), tokiose kaip personaliniai kompiuteriai, PLC valdikliai ir pan.

RISC (Reduced Instruction Set Computing) – tai mikrovaldiklio architektūros (objekto elgsenos aprašo) tipas, kuris naudoja mažas, tačiau optimizuotas instrukcijas mikrovaldiklio funkcijoms atlikti.

ADC (Analogue to Digital Converter) keitiklis – tai įtaisas, kuris konvertuoja analoginį signalą į skaitmeninį.

Anemometras – prietaisas, skirtas vėjo srauto greičiui matuoti.

Kritinis vėjo greitis – pavojingas vėjo greitis, kuris gali pažeisti vėjo jėgainės konstrukcines dalis (mentys, generatorių ir t.t.).

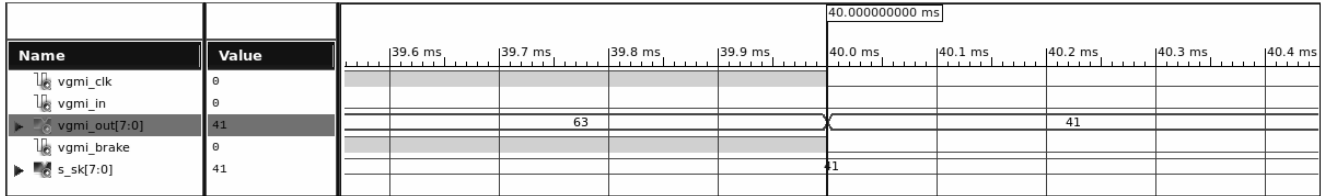
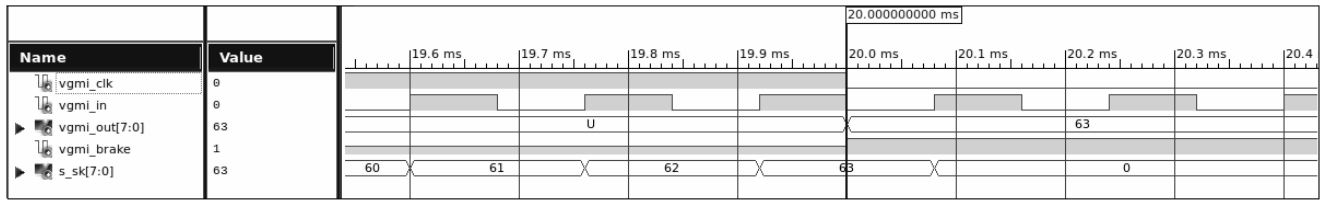
PID reguliatorius – parametriškai optimizuojamas reguliatorius, kurį sudaro 3 grandys: proporcinė (P), integruojanti (I) ir diferencijuojanti (D).

SCADA (Supervisory Control and Data Acquisition) – tai grafinė PLC valdiklių sąsaja (vizualizacinė sistema), kuri dažniausiai naudojama pramoninių automatizuotų sistemų nuotoliniam (pvz. iš operatoriaus darbo vietos personalinio kompiuterio) valdymui, būsenų stebėjimui įvykių, archyvinių duomenų analizavimui.

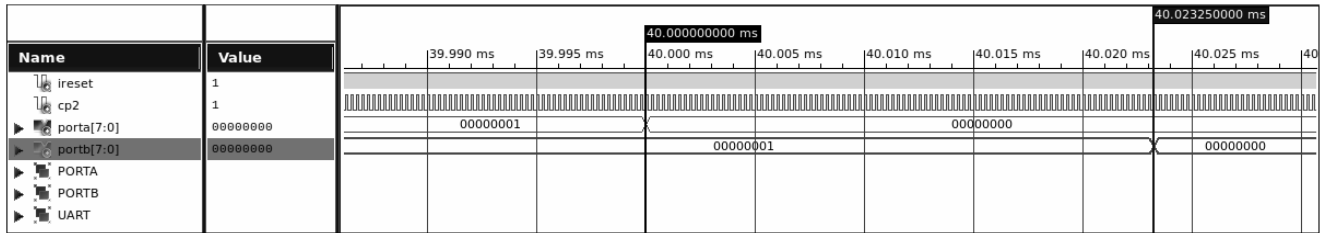
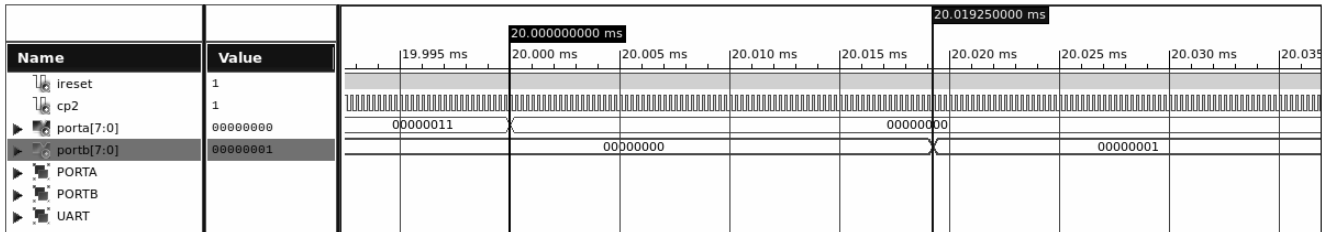
LITERATŪRA

1. Charles Sevier 2009, *Wind Turbine Control*, p. 5-10.
2. Marius Constantin Popescu, Mircea Grigoriu, Doina Cornelia Dinu, *Wind Turbine Testing Control System*, 2008, p. 168-171.
3. Dr. Horizon Gitano – Briggs, 2010, *Small Wind Turbine Power Controllers*, p. 165-188.
4. Kompanijos Bonus Energy straipsnis, 1998, *The Wind Turbine Components and Operation*, p. 20-24.
5. William E. Leithead, Sergio Dominguez, 2005, *Coordinated Control Design for Wind Turbine Control Systems*, p. 1-8.
6. Gytis Petrauskas, 2000, *Modernių vėjo jėgainių savybės ir plėtojimo perspektyvos*, p. 74-78.
7. EngineerIT, Information from National Instruments, *Combining PLC and FPGA architectures*, 2006, p. 64-67.
8. Kompanijos Xilinx techninė dokumentacija, 2011, *Spartan-6 Family Overview*, p. 1-11.
9. Clive Maxfield, EE Times Design, 2006, *FPGA Architecture „A“ to „Z“ Part 1, Part 2*“
prieiga per internetą: <http://www.eetimes.com/design/programmable-logic>.
10. Kompanijos Xilinx techninė dokumentacija, 2010, *Spartan-3 Generation FPGA User Guide*,
p. 65-214, p. 315-373.
11. Bob Zeidman, 2003, *Introduction to CPLD and FPGA Design*, p. 9-15.
12. Ian Koun, Russell Tessier ir Jonathan Rose, 2007, *FPGA Architecture: Survey and Challenges*,
p. 136-150, p. 176-198.
13. Claus Kuhnel, 1990, *AVR RISC Microcontroller Handbook*, p. 1-34.
14. Dr. Peter R. Wilson, 2006, *Design Recipes for FPGA*, p. 5-41.
15. Kompanijos Xilinx techninė dokumentacija, 2011, *Spartan-6 FPGA Data Sheet*, 1-76.
16. Pong P. Chu, 2007, *FPGA Prototyping by VHDL Examples*, p. 11-17.
17. V. Bulovas, 1989, *Mikroprocesoriai*, p. 25-32.
18. Atmel kompanijos techninė dokumentacija, rev. 0945I, 2007, *Microcontroller with 128K Bytes
In-System Programmable Flash, ATmega103*, p. 3-122.
19. Designing Controllers for a Doubly-Fed Wind Power System, 2011, prieiga per internetą:
<http://zone.ni.com/devzone/cda/epd/p/id/6272>.
20. Švietimo ir mokslo ministerija, Mokslo ir technologijų populiarinimo projektas „Apie energiją
mąstyk kitaip“, 2008, *Atsinaujinantys energijos šaltiniai*, p. 1-6.
21. Adolf Thies techninė specifikacija, 2005, *Small Wind Transmitter*, p. 3-6.
22. M. Regheb, 2009, Control of wind turtines, p. 1-14

PRIEDAI



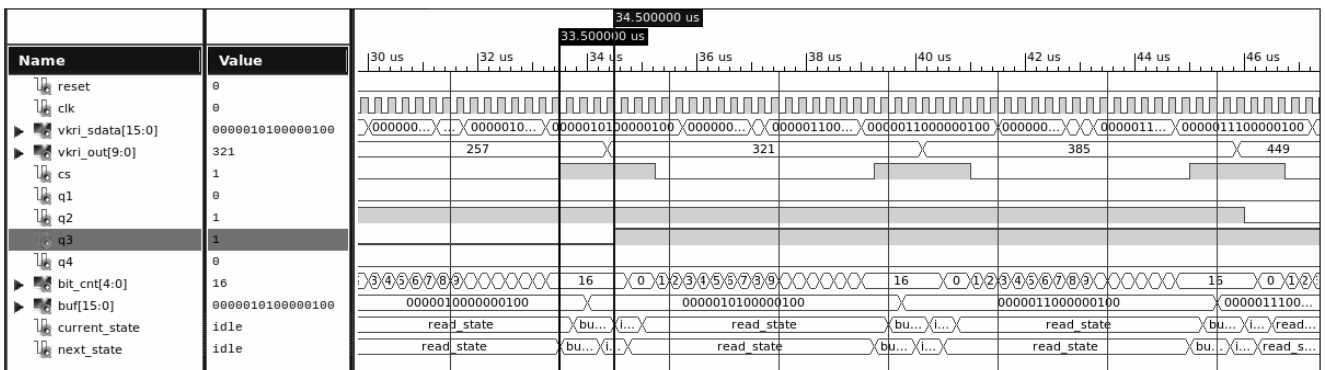
a)



b)

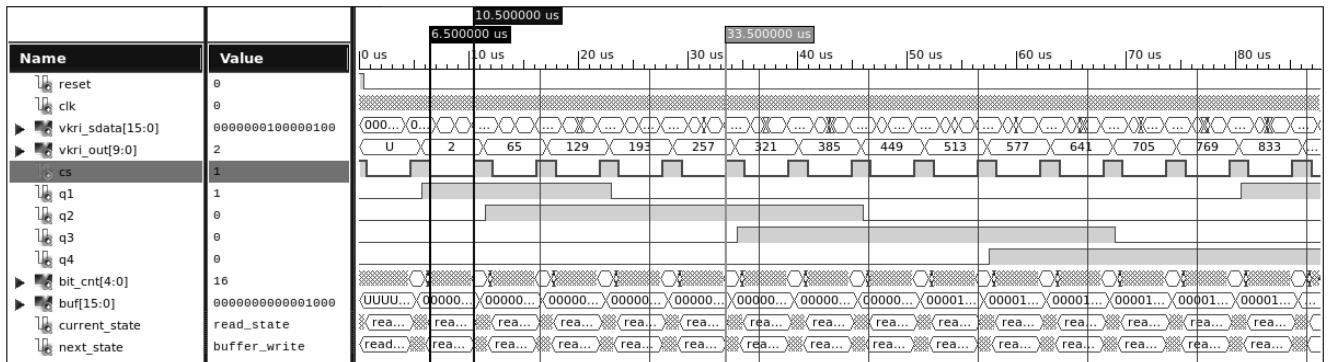
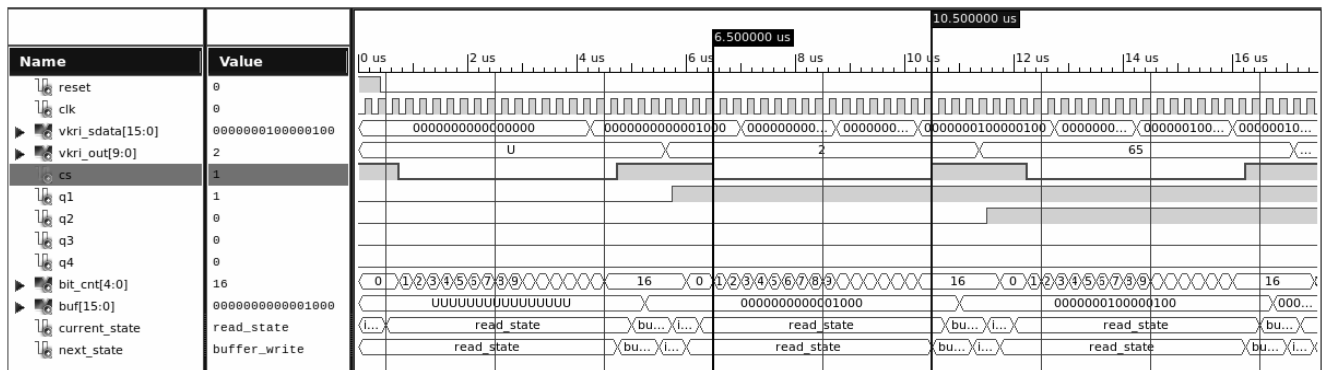
1 pav. Modeliavimo rezultatai, kur a) Vėjo jėginių valdymo sistemos Nr.: 1³ VGMĮ algoritmo funkcijų atlikimas laiko atžvilgiu; b) Vėjo jėginių valdymo sistemos Nr.: 2⁴ VGMĮ algoritmo funkcijų atlikimas laiko atžvilgiu

Šaltinis: sudaryta autoriaus

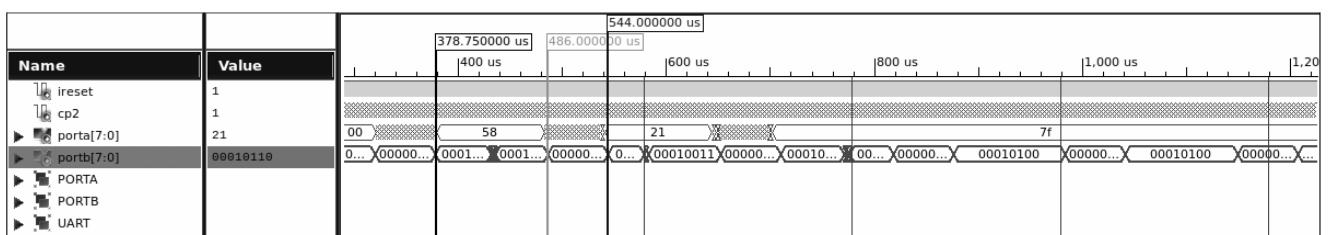
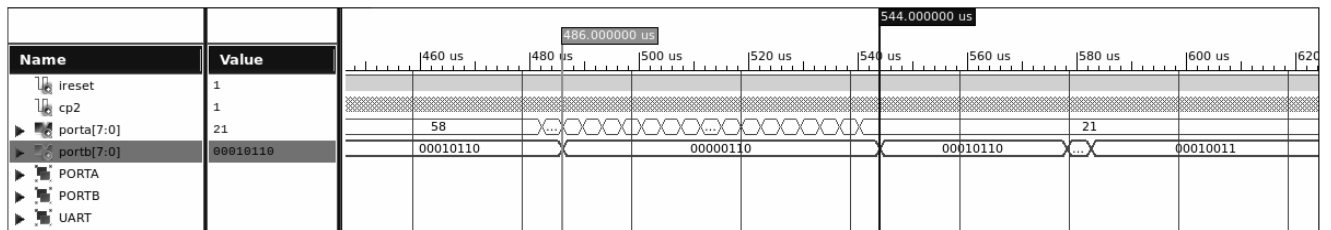
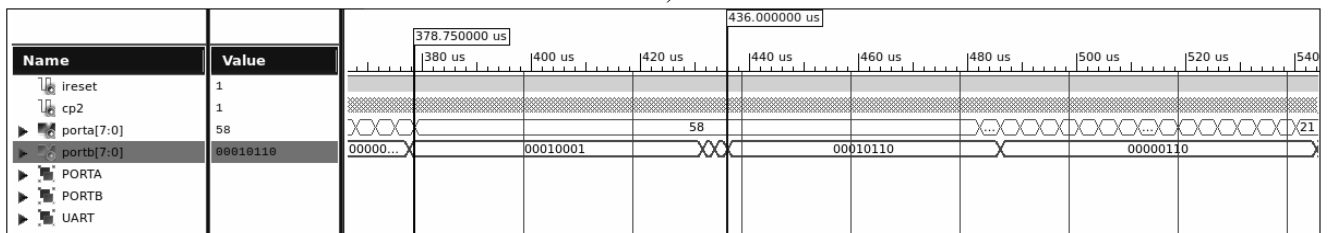


³ Vėjo jėginių valdymo sistemos Nr.: 1 – tiesiogiai FPGA matricoje realizuotas valdymo algoritmas.

⁴ Vėjo jėginių valdymo sistemos Nr.: 2 – MCU RISC (Atmega103) architektūros pagrindu realizuotas valdymo algoritmas.



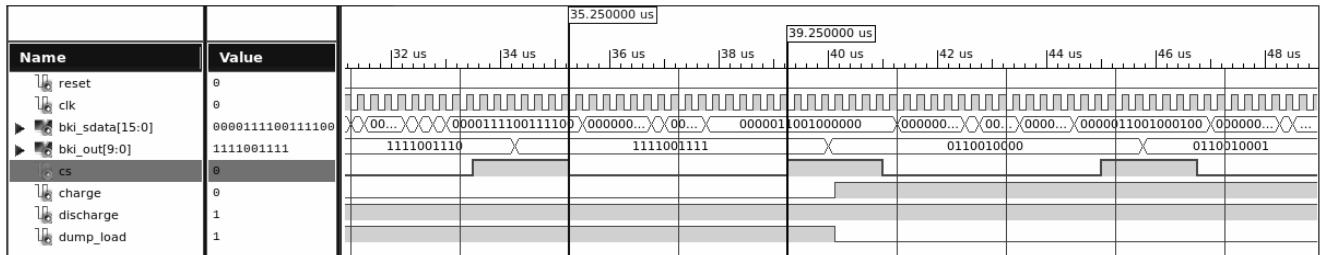
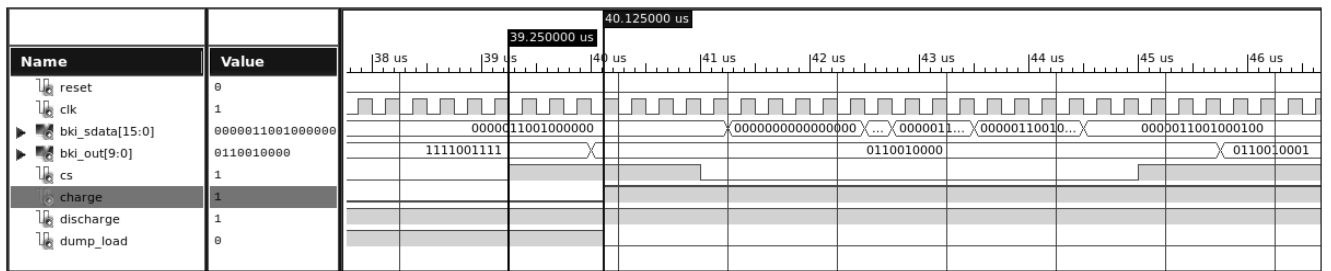
a)



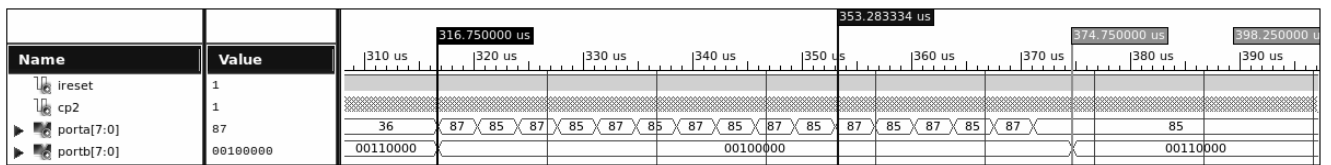
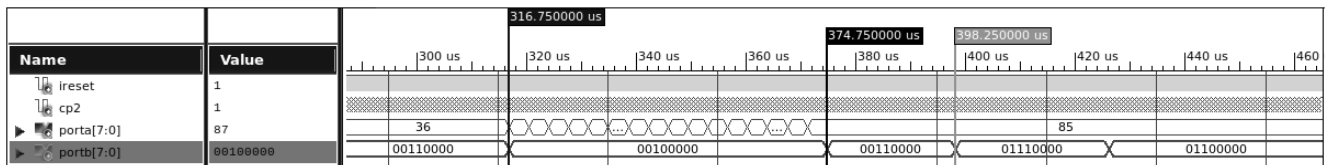
b)

2 pav. Modeliavimo rezultatai, kur a) Vėjo jėginių valdymo sistemos Nr.: 1 VQRĮ algoritmo funkcijų atlikimas laiko atžvilgiu; b) Vėjo jėginių valdymo sistemos Nr.: 2 VQRĮ algoritmo funkcijų atlikimas laiko atžvilgiu

Šaltinis: sudaryta autoriaus



a)



b)

3 pav. Modeliavimo rezultatai, kur a) Vėjo jėginių valdymo sistemos Nr.: 1 BKĮ algoritmo funkcijų atlikimas laiko atžvilgiu; b) Vėjo jėginių valdymo sistemos Nr.: 2 BKĮ algoritmo funkcijų atlikimas laiko atžvilgiu

Šaltinis: sudaryta autoriaus

Aukščiau matyti keletą algoritmų modeliavimo rezultatų, kuriais remiantis sudarytas sistemų tarpusavio palyginimas. Rezultatai pateikti poskyrio „2.2.3.3 Realizuotų vėjo jėginės valdymo sistemų rezultatų palyginimas“ lentelėse.