

Electronics and electrical engineering  
Elektronika ir elektros inžinerija

KOMPIUTERIŲ ARCHITEKTŪROS UŽDUOČIŲ SUDARYMAS TYRIMAMS,  
ATLIEKAMIEMS LAUKU PROGRAMUOJAMOMIS LOGINĖMIS  
MATRICOMIS

Kęstutis BARTNYKAS \*

*Vilniaus Gedimino technikos universitetas, Vilnius, Lietuva*

Gauta 2021 m. birželio 22 d.; priimta 2021 m. birželio 30 d.

**Santrauka.** Kompiuterių architektūros užduotys dažnai skirtos tyrimams atlikti lauku programuojamomis loginėmis matricomis. Studentas pagal pateiktą specifikaciją turi nurodyta HDL (angl. *hardware description language*) kalba aprašyti procesorių su jo veikimui būtiniais išoriniais komponentais. Problema yra ta, kad šiuo atveju pagrindas yra vienos konkrečios architektūros procesorius, todėl dėstytojas susiduria su užduočių individualizavimo problema. Šiame straipsnyje siūlomas sprendimas, kurio pagrindas yra skirtingi procesoriai vietoje vieno konkrečios architektūros programuojamo procesoriaus. Čia parodoma, kad taip žiūrint užduočių individualizavimo problema lengviau sprendžiama, o nuo kompiuterių architektūros teorijos nenu-  
tolstama, nes programuojamas procesorius yra apibendrintas skirtingo procesoriaus atvejis. Straipsnyje pateikti užduočių pa-  
vyzdžiai skirti tyrimams atlikti lauku programuojamomis loginėmis matricomis. Pateiktos sąsajos su kompiuterių architek-  
tūros teorine medžiaga. Aprašomas kitas užduočių variantas negu buvo taikytas praktikoje dėstant kompiuterių architektūros  
studijų modulį VILNIUS TECH Elektroninių sistemų katedroje, t. y. atlikti tam tikri pakeitimai ir papildymai.

**Reikšminiai žodžiai:** kompiuterių architektūra, užduotis, tyrimas, modeliavimas, VHDL sintaksė, lauku programuojamos loginės matricos.

## Įvadas

Pažymėtina, kad kompiuterių architektūroje yra nagri-  
nėjama skaitmeninių kompiuterių sąsaja tarp aparatinės  
įrangos ir programinės įrangos. Šis straipsnis yra kom-  
piuterių architektūros tema, tačiau atkreiptinas dėmesys  
į tai, kad žvelgiant į skirtingus literatūros šaltinius matyti  
skirtingos šios temos interpretacijos. Pavyzdžiui, vienoje  
knygoje kompiuterių architektūra aiškinama žvelgiant iš  
aparatinės įrangos pusės, o kitose iš programinės įrangos  
pusės (Comer, 2020). Taip žvelgiant šis straipsnis gali pa-  
sirodyti dar viena interpretacija kompiuterių architektūros  
tematika.

Kompiuterių architektūros studijų modulyje dažnai  
naudojamas užduotys, skirtos tyrimams atlikti lauku pro-  
gramuojamomis loginėmis matricomis (angl. *field pro-  
grammable gate array, FPGA*) (Strelzoff, 2007; Park et al.,  
2013; Zavala et al., 2015; McGrew et al., 2019). Dažnai tai  
reiškia, kad pasirenkamas vienas konkrečios architektūros  
procesorius ir įgyvendinamas lauku programuojamomis  
loginėmis matricomis. Šiuo atveju studentui pateikiama  
procesoriaus techninė specifikacija, pagal kurią studentas

nurodyta HDL kalba aprašo procesorių su jo veikimui  
būtiniais išoriniais komponentais. Dažniausiai tai būna  
VHDL (angl. *very high speed integrated circuit hardware  
description language*) arba Verilog sintaksė. Taip studentas,  
atlikdamas užduotis, susipažįsta su procesoriaus architek-  
tūra, jo veikimui užtikrinti būtiniais išoriniais komponen-  
tais. Atlikdamas užduotis studentas taip pat susipažįsta  
su procesoriaus modeliavimu – aprašo procesorių, jį per-  
kelia į lauku programuojamų matricų maketą ir išban-  
do praktiškai. Iš esmės darbo rezultatas yra HDL kalba  
parašytų failų rinkinys. Kitų literatūroje aprašomų būdų  
mokyti kompiuterių architektūros pagrindas yra tik mo-  
deliavimas, t. y. lauku programuojamų matricų maketas  
nenaudojamas. Bet kuriuo atveju problemos, su kurio-  
mis susiduria dėstytojas, neaprašomos. Tokia problema  
yra studentų užduočių individualizavimas, kuris padėtų  
užtikrinti aukštosios mokyklos etikos kodekso laikymą-  
si. Problema yra ta, kad pasirinkus tik vienos konkrečios  
architektūros procesorių prasmingas užduočių individua-  
lizavimas tampa problematiškas.

\*Autorius susirašinėti. El. paštas [kestutis.bartnykas@vilniustech.lt](mailto:kestutis.bartnykas@vilniustech.lt)

Šiame straipsnyje siūlomas sprendimas, kurio pagrindas yra skirtiniai procesoriai (angl. *dedicated processors*) vietoje vieno konkrečios architektūros programuojamo procesoriaus (angl. *general purpose processor*). Standžiojo disko valdiklis, tinklo plokštė, vaizdo plokštė yra tokių procesorių pavyzdžiai. Galimi ir žymiai paprastesni skirtinių procesorių pavyzdžiai. Šiame straipsnyje pateikiama tokių pavyzdžių. Pabrėžtina, kad programuojamas procesorius yra skirtingų procesorių apibendrinantis atvejis. Pasirinkus skirtingų procesorių kaip sprendimo būdą, prasmingas užduočių individualizavimas tampa paprastesnis. Dar vienas siūlomo sprendimo būdo privalumas yra tas, kad gali būti aprėpiama daugiau temų. Pavyzdžiui, kai užduotys formuojamos vienos konkrečios architektūros procesoriaus pagrindu, apsiribojama tik viena technologija, pavyzdžiui RISC (angl. *reduced instruction set computer*) technologija. Čia pateikiamas sprendimas yra lankstesnis. Dar vienas dalykas, į ką verta atkreipti dėmesį, yra tas, kad moksliniuose straipsniuose dažniausiai aprašomu atveju studento sukurtas procesorius, atlikus dėstytojo nurodytas užduotis, kituose studijų moduluose gali būti panaudotas tik tada, kai jam taip pat buvo sukurti kompiliatoriai ir saistykėlė, kurie geba dirbti su šiaudien aktualiomis aukšto lygio programavimo kalbomis. Kitu atveju užtikrinti pradėto darbo tęstinumą neįmanoma. Skirtinis procesorius pradėto darbo tęstinumui nėra kliūtis.

## 1. Literatūros apžvalga

Kai kuriuose moksliniuose straipsniuose, skirtuose kompiuterių architektūrai mokytis, pagrindinis dėmesys skiriamas procesoriui ir susitelkiama į procesoriaus modeliavimą taip kaip Lee et al. (2012), Zavala et al. (2015), Ichsan ir Kurniawan (2017), Schuurman (2013) straipsniuose. Kartais vietoje procesoriaus modeliavimo taikant HDL naudojama konkrečiam procesoriui skirta modeliavimo programa. Pirmuoju atveju studentas, modeliuodamas procesorių, mokosi dirbti su HDL, antruoju atveju – pažintis su procesoriumi yra visiškai teorinė. Vis dėlto abiem atvejais siekiama surasti naują būdą perteikti studijuojamą medžiagą ir padidinti studento motyvaciją suvokti gana abstrakčias sąvokas.

Kituose straipsniuose, be naujų mokymo būdų, taip pat siekiama užtikrinti studijų nuoseklumą. Strelzoff (2007) atkreipia dėmesį, kad atliekant kompiuterių architektūros studijų modulio užduotis HDL kalba sumodeliuoti procesoriai galėtų būti panaudoti kaip įterptiniai procesoriai kituose studijų moduluose, tačiau lieka nepanaudoti. Priežastis ta, kad modeliuojami procesoriai neturi savo kompiliatoriaus ir saistykėlės, kuriais būtų galima paruošti šiaudien aktualia aukšto lygio programavimo kalba parašytą kodą vykdyti tokiaame procesoriuje. Šių programinių įrankių projektavimas nėra kompiuterių architektūros objektas, dėl to Strelzoff siūlo mokymui naudoti procesorių modelius (angl. *soft processors*), aprašytus HDL kalba, kurie teikiami kartu su FPGA sintezavimo įrankiais. Autorius kalba apie VHDL sintaksę. Šių procesorių komandų sistemos sutampa su kai kurių gerai žinomų programuoja-

mų procesorių komandų sistemomis. Tai reiškia, kad darbai su tokiais procesoriais reikalingi kompiliatoriai ir saistykėlės yra sukurti. Taip pat galima sakyti, kad šiuo atveju studijų modulyje studijuojama šiaudien aktuali medžiaga. Straipsnio autorius pateikia užduočių įvairiomis temomis – aktualesni pavyzdžiai atsižvelgiant į šį laikmetį yra tokie. Kai kuriose užduotyse procesoriaus modelį siūloma panaudoti kaip valdiklį, kuriuo įgyvendinamas konkretus spartinimo būdas (angl. *caching*), būdingas masiškai gaminamiems programuojamiems procesoriams. Kitose užduotyse siūloma iš keleto procesorių modelių sumodeliuoti vaizdo plokštę, kuri galėtų vykdyti vektorines operacijas (angl. *single instruction multiple data instructions, SIMD*). Kitaip sakant, siūloma sukurti sudėtingesnę procesorių panaudojant jau turimus paprastesnius. Kituose savo esme panašiuose straipsniuose (Park et al., 2013) atkreipiamas dėmesys į tai, kad gamintojų, užsiimančių lustų gamyba, didelę dalį produkcijos sudaro daugiabranduoliai procesoriai. Be to, daugiabranduoliai procesoriai yra gana plačiai paplitę įvairiose srityse, pavyzdžiui, – mobilieji telefonai. Čia aprašomų užduočių esmė yra dviejų arba daugiau MIPS (angl. *microprocessor without interlocked pipeline stages*) procesorių sąveikos užtikrinimas pritaikant bendrą atmintį. Tam studentas turi HDL kalba sumodeliuoti šią atmintį valdantį valdiklį. Autorius pasirinkęs *Verilog* sintaksę. Tada sumodeliuotas daugiabranduolis procesorius panaudojamas įvykdyti studento parašytą programą, kuri rašoma taikant lygiagrečiojo programavimo principus. Pagal tematiką panašus darbas yra Udugama et al. (2013). Čia aprašomas kompiuterių architektūros ir lygiagrečiojo programavimo principams mokytis skirtas daugiabranduolis procesorius. Šių procesorių sudaro šešiolika SEP (*students' experimental processor*) branduolių, kurie yra sujungti panaudojant dvimatį tinklą, ir jis yra sumodeliuotas panaudojant VHDL sintaksę. Jį galima pritaikyti įvairioms lygiagrečiosioms architektūroms, aprašomoms Flynno taksonomijoje, modeliuoti. Dar vienas straipsnis, skirtas tam tikriems įgūdžiams suformuoti, paskelbtas Markettos et al. (2016). Čia išskeltas tikslas aprėpti kuo daugiau technologijų, t. y. HDL, FPGA, Yarvi RISC-V procesorius, assembleris, programavimas C kalba, programavimas pradedant kodo rašymą nuo testų rašymo (angl. *test-driven development*). Panašiais atributais pasižymi Nisan ir Schocken (2008) darbas. Čia išvardytais atvejais tikslas yra ne tik studijuojančio asmens motyvacijos skatinimas ir sąvokų suvokimas, tačiau šiaudieninis aktualumas ir tam tikrų įgūdžių formavimas.

Atkreipkime dėmesį į šiuos dalykus, kylančius iš straipsnių apžvalgos. Įprastu atveju, norint užtikrinti tęstinumą, reikia pasirūpinti kompiliatoriumi ir saistykle, kurie dirbtų su šiaudien aktualiomis aukšto lygio programavimo kalbomis – skirtiniam procesoriui nereikalinga programinė įranga, nes programinė įranga „įrašyta“ jame. Kitaip sakant, modeliuojami procesoriai dažnai neturi čia paminėtos programinės įrangos, todėl nėra tęstinumo. Apžvelgtuose straipsniuose aprašomas vienas ir tas pats darbas, skirtas visiems studentams, t. y. galimybė individualizuoti užduotis neaptariama.

## 2. Sprendžiama problema

Šiame straipsnyje keliamas tikslas – parodyti, kaip gali būti sukurtos kompiuterių architektūros užduotys tyrimams, atliekamiems lauku programuojamomis matricomis, tyrimo objektu pasirinkus skirtingus procesorius.

Pagrindiniai reikalavimai, kurių buvo siekiama formuojant užduotis tyrimams:

- Užduotyje numatytas tyrimas turi būti susijęs su kompiuterių architektūra.
- Užduotys turi būti nesudėtingai individualizuojamos.
- Užduotys turi leisti formuoti praktinius įgūdžius dirbant su VHDL sintakse ir lauku programuojamomis loginėmis matricomis.

Straipsnyje daroma prielaida, kad kompiuterių architektūros užduotis galima lengviau individualizuoti kaip pagrindą užduotims formuoti pasirenkant ne vienos konkrečios architektūros programuojamą procesorių, bet skirtingus procesorius.

## 3. Atliekami tyrimai

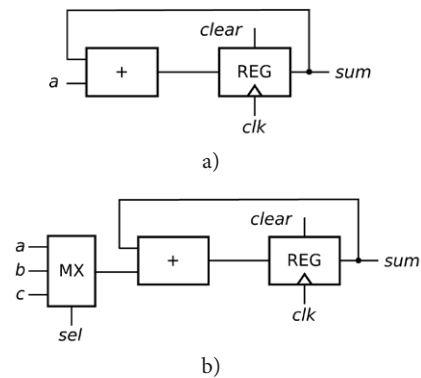
Toliau aprašomos užduočių temos ir paaiškinamas kiekviena tema atliekamas tyrimas. Tyrimų esmė – aritmetinio reiškinio modeliavimas, todėl individualizavimas atliekamas labai paprastai. Kadangi privaloti įvairių aritmetinių reiškinų nesudėtinga, todėl nesudėtinga ir sukurti individualias užduotis. Žinoma, aritmetinis reiškinys turi būti sumodeliuotas taikant tam tikrus principus, o ne tiesiog įgyvendintas pritaikant VHDL sintaksės galimybes. Modeliuojant aritmetinį reiškinį įgyvendinančią loginę grandinę reikia sukurti duomenų apdorojimo traktą ir jam valdyti skirtą valdiklį. Tada galima pritaikyti kitus papildomus principus. Straipsnyje aprašomas konvejerinis apdorojimas.

Studijų modulis *kompiuterių architektūra* studijuojamas VILNIUS TECH Elektroninių sistemų katedroje. Tai vieno semestro trukmės studijų modulis, kuriam reikia penkių užduočių tyrimams. Studijuodami kompiuterių architektūrą studentai nebūna susipažinę nei su HDL kalbomis, nei VHDL sintakse, kuri vartojama atliekant čia aprašomus tyrimus, todėl kaip pagrindinė medžiaga rekomenduojama Tappero ir Mealy (2013) knyga, kurioje gausiais pavyzdžiais paaiškinta VHDL sintaksė. Kito semestro metu ši sintaksė vartojama atliekant projektą, skirtą kompiuterio elementams projektuoti. Studentams pateikiamos užduotys suformuluotos taip, kad atliekamo tyrimo esmė būtų loginės grandinės modeliavimas ir patikrinimas, ar ji teisinga, bei aprašyto modelio perkėlimas į lauku programuojamas logines matricas.

Pirmoji užduotis skirta atlikti tyrimui, kurio esmė yra susipažinimas su darbo aplinka. Omenyje turima programinė įranga *Altera Quartus II*. Tyrimo tikslas yra susipažinti su pagrindine VHDL sintakse ir modeliuoti skirtingais įrankiais. Tai programa *Modelsim*, kuri yra *Altera Quartus II* programinės įrangos dalis. Akcentas yra modeliavimas, susipažinimas su modeliavimo procesu. Tam studentas sumodeliuoja labai paprastą loginę grandinę taikydamas paprasčiausią VHDL sintaksę, t. y. logines išraiškas.

Antroji užduotis yra pirmosios tęsinys, t. y. toliau tęsiama pažintis su VHDL sintakse, tačiau tyrimo tikslas yra susipažinti su sudėtingesne VHDL sintakse, o jos mokymasis tęsiamas ir per kitus tyrimus. Per tam skirtą laiką susipažįstama su tokiais pagrindiniais dalykais kaip sakiniai *with / select*, *when / else* bei *if* ir *case*.

Trečioji užduotis skirta atlikti tyrimui su logine grandine, ji skirta apdoroti duomenims, t. y. duomenų apdorojimo traktu. Tokios grandinės pavyzdžių pateikta 1 paveiksle. Šio paveikslo a dalyje pavaizduota grandinė, sumuojanti kintamojo  $a$  reikšmę  $n$  kartų. To paties paveikslo b dalyje pavaizduota grandinė, kuri susumuoja įėjimo kintamuosius  $a$ ,  $b$  ir  $c$ . Ši tema sudaro sąlygas suprasti, kad įrašyti duomenis į registrą ir juos nuskaityti per tą patį sinchronizavimo signalo ciklą yra neįmanoma. Tai gali būti naudinga suvokiant teorinę medžiagą, pavyzdžiui, aptariant komandų konvejerio veikimo principus ir pan.



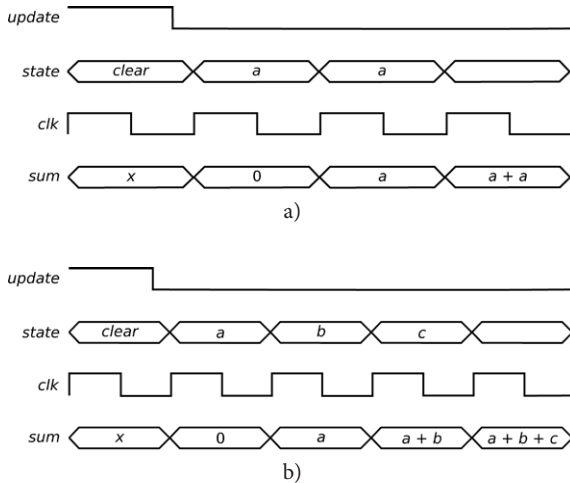
1 paveikslas. Grandinių pavyzdžiai, kurios modeliuojamos atliekant tyrimą pagal trečią užduotį: a – grandinė, sumuojanti kintamojo  $a$  reikšmę  $n$  kartų; b – grandinė, sumuojanti kintamuosius  $a$ ,  $b$  ir  $c$

Figure 1. Examples of datapath circuits, that should be simulated according to the third problem statement:

- a – datapath circuit summing up to  $n$  times input variable  $a$ ;
- b – datapath circuit responsible for summation of three variables  $a$ ,  $b$  and  $c$

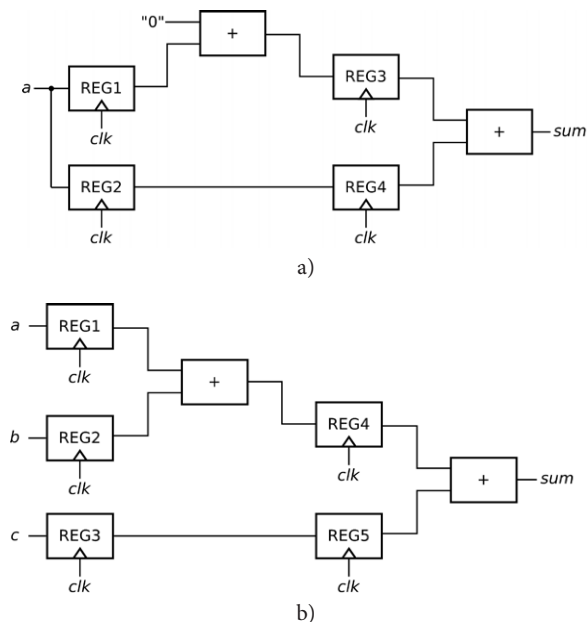
Ketvirtoji užduotis skirta atlikti tyrimui su valdikliu. Atlikdamas šį tyrimą studentas sukuria valdiklį. Čia omenyje turimas valdiklis, kuris skirtas anksčiau sukurtai grandinei valdyti siekiant apdoroti duomenis. Jį įprasta aprašyti būvių automatu. Projektuojant būvių automatą nurodoma, iš kurio registro į kurį bus perduodami duomenys ir kuriuo metu tai turi būti padaryta. Valdiklis turi įėjimus ir išėjimus. Atliekant šį tyrimą valdiklio išėjimo signalai skirti valdyti tokiems komponentams kaip, pavyzdžiui, multiplekseriai, o valdiklio įėjimo signalais atnaujinamas valdiklio būvis, pavyzdžiui, pasikeitus reikšmėms duomenis apdorojančios grandinės įėjime. Laiko diagramos, vaizduojančios, kaip duomenis apdorojanti grandinė turi veikti, pateiktos 2 paveiksle. Šio paveikslo a dalyje pateiktos laiko diagramos grandinei iš 1 paveikslo a dalies, o b dalyje laiko diagramos grandinei iš 1 paveikslo b dalies. Šio tyrimo sąsajos su teorine medžiaga yra tokios: susipažįstama su valdikliu, valdymo žodžiu, valdymo žodį sudarančiais valdymo bitais ir pan. Viena iš apibendrintos

procesoriaus struktūros sudedamųjų dalių yra valdymo įtaisas. Vis dėlto valdymo įtaiso sudedamoji dalis yra valdiklis, kuris kartu su specialiais registrais atlieka procesoriaus valdymo funkciją.



2 paveikslas. Valdiklio veikimą aprašančios laiko diagramos grandinėms, kurias reikia sumodeliuoti atliekant tyrimą pagal ketvirtą užduotį: a – diagrama grandinei iš 1 paveikslą a dalies; b – diagrama grandinei iš 1 paveikslą b dalies

Figure 2. Timing diagrams for controllers, that should be simulated according to the fourth problem statement: a – timing diagram for the datapath circuit in Figure 1, part a; b – timing diagram for the datapath circuit in Figure 1, part b



3 paveikslas. Grandinių, kurios modeliuojamos atliekant tyrimą pagal penktą užduotį, pavyzdžiai. Tai ankstesnio tyrimo metu sumodeliuotos grandinės, perdarytos pritaikant konvejerio principus: a – diagrama grandinei iš 1 paveikslą a dalies; b – diagrama grandinei iš 1 paveikslą b dalies

Figure 3. Datapath circuits with pipelining principles applied that should be simulated according to the fifth problem statement: a – circuit for the datapath circuit in Figure 1, part a; b – circuit for the datapath circuit in Figure 1, part b

Penktoji užduotis skirta atlikti tyrimui su konvejeriu. Tyrimo esmė yra labai paprasta: tam tikrose nagrinėjamos skaitmeninės grandinės vietose reikia sudėti papildomus registrus (angl. *pipeline latch*). Taip perdaryta grandinė, kai prisipildo duomenų, skaičiavimo rezultatą duoda kas vieną ciklą. Pradinė grandinė turi turėti bent du funkcinius įtaisos. Tokių grandinių pavyzdžiai pateikti 3 paveiksle. Šio tyrimo metu studentas grandinę, sukurtą ankstesnio tyrimo metu, perdaro taip, kad jos veikimas atitiktų konvejerinio apdorojimo principus. 3 paveikslo a dalyje pateiktas 1 paveikslo a dalies atitikmuo, o to paties paveikslo b dalyje pateiktas 1 paveikslo b dalies atitikmuo. Tokio tyrimo atlikimas šia tema svarbus dėl šios priežasties. Konvejeris taikomas ne tik apdorojant komandas. Konvejerio veikimo principai gali būti pritaikyti įvairioms loginėms grandinėms.

## Išvados

Apibendrinant šiame straipsnyje iškeltą problemą galima daryti tokias išvadas:

- Skirtinių procesorių tema kaip pagrindas formuluojant kompiuterių architektūros užduotis tyrimams, atliekamiems lauku programuojamomis matricomis, padeda paaiškinti ir aptarti kai kuriuos terminus, būdingus kompiuterių architektūrai.
- Čia aprašytas kompiuterių architektūros užduočių individualizavimo problemos sprendimo būdas leidžia nesudėtingai ir prasmingai individualizuoti užduotis įvairiomis temomis.
- Pasirinktas sprendimo būdas leidžia formuoti praktinius įgūdžius, tokius kaip loginių grandinių aprašymas VHDL sintakse, modeliavimas, darbas su FPGA maketu.

Verta paminėti, kad užduotyse nurodytiems tyrimams galima suteikti daugiau unikalumo kitų studijų modulių atžvilgiu testus rašant VHDL sintakse arba TCL (angl. *tool command language*) skriptais vietoje rankinio testavimo. Nemaža dalis programinių įrankių, skirtų darbui su VHDL sintakse, nesvarbu, ar tai būtų modeliavimas, ar sintezavimas, naudoja TCL technologiją. Tai leidžia rašyti TCL skriptus, kurie atliekant modeliavimą sąveikauja su VHDL sintakse parašytu kodu. Taip pat tai suteikia galimybę VHDL sintakse parašytą kodą panaudoti pakartotinai tiek skirtinguose programiniuose įrankiuose, tiek skirtingose platformose. Taip studentai būtų supažindinti, kaip parašyti testą taikant VHDL sintaksę arba TCL skriptą.

## Literatūra

- Comer, D. (2020). *Essentials of computer architecture* (2nd ed.). Taylor & Francis.
- Ichsan, M. H. H., & Kurniawan, W. (2017). Design and implementation 8 bit CPU architecture on Logisim for undergraduate learning support. In *2017 International Conference on Sustainable Information Engineering and Technology (SIET)* (pp. 132–137). IEEE. <https://doi.org/10.1109/SIET.2017.8304123>
- Lee, J. H., Lee, S. E., Yu, H. C., & Suh, T. (2012). Pipelined CPU design with FPGA in teaching computer architecture. *IEEE Transactions on Education*, 55(3), 341–348. <https://doi.org/10.1109/TE.2011.2175227>

- Markettos, A. T., Moore, S. W., Jones, B. D., Spliet, R., & Gavri-la, V. A. (2016). Conquering the complexity mountain: full-stack computer architecture teaching with FPGAs. In *2016 11th European Workshop on Microelectronics Education (EWME)* (pp. 1–6). IEEE. <https://doi.org/10.1109/EWME.2016.7496457>
- McGrew, T., Schonauer, E., & Jamieson, P. (2019). Framework and tools for undergraduates designing RISC-V processors on an FPGA in computer architecture education. In *2019 International Conference on Computational Science and Computational Intelligence* (pp. 778–781). IEEE. <https://doi.org/10.1109/CSCI49370.2019.00148>
- Nisan, N., & Schocken, S. (2008). *The elements of computing systems: building a modern computer from first principles*. MIT Press.
- Park, H., Ko, Y.-W., So, J., & Lee, J.-G. (2013). Synthesizable manycore designs with FPGA in teaching computer architecture. *International Journal of Control and Automation*, 6(5), 429–438. <https://doi.org/10.14257/ijca.2013.6.5.38>
- Schuurman, D. C. (2013). Step-by-step design and simulation of a simple CPU architecture. In *SIGCSE 2013: Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (pp. 335–340). ACM. <https://doi.org/10.1145/2445196.2445296>
- Strelzoff, A. (2007). *Teaching computer architecture with FPGA soft processors* [Conference presentation]. American Society for Engineering Education (ASEE) Southeast Section Conference.
- Tappero, F., & Mealy, B. (2013). *Free range VHDL. The no-frills guide to writing powerful code for your digital implementations*. Free Range Factory.
- Udugama, L. S. K., Geeganage, J., & Kuruppuarachchi, W. V. (2013). A configurable multi-core processor for teaching parallel processing. In *2013 IEEE 8th International Conference on Industrial and Information Systems* (pp. 326–331). IEEE. <https://doi.org/10.1109/ICIInfS.2013.6732004>
- Zavala, A. H., Huerta-Ruelas, J. A., Camacho, O., & Carvallo-Domínguez, A. R. (2015). Design of a general purpose 8-bit RISC processor for computer architecture learning. *Computación y sistemas*, 19(2), 371–385. <https://doi.org/10.13053/cys-19-2-1941>

## PROJECT DESIGN FOR COMPUTER ARCHITECTURE PRACTICAL SESSIONS BASED ON FIELD-PROGRAMMABLE GATE ARRAY

K. Bartnykas

Abstract

Field-programmable logic arrays are often used in courses on computer architecture. The student must describe the processor with the external components necessary for its operation in the specified HDL (hardware description language) language according to the provided specification during a certain number of projects. The weakness of this approach is that the basis of such projects is a processor of one specific architecture, so the lecturer faces the issue of individualization of projects. This article proposes a solution based on dedicated processors instead of one programmable processor of a specific architecture. It's shown here that the issue of project individualization is easier solvable in the proposed way, and it does not deviate from the theory of computer architecture, because the programmable processor is a generalization of a dedicated processor. The article describes project design ideas based on dedicated processors and gives some examples. Represented different instance than was applied during practical sessions of Computer Architecture that are held at the Department of Electronic Systems within VILNIUS TECH, i.e. certain modifications, and additions were applied.

**Keywords:** computer architecture, project design, simulation, VHDL syntax, field programmable gate array.