

Vilniaus universitetas

Fizikos fakultetas

Skaitmeninės elektronikos laboratorija

Laboratorinis darbas nr. 9

Loginių grandinių tyrimas

Užduotis

1. Sudaryti duotos skaitmeninės grandinės (duoda dėstytojas) reikšmių lentelę ir užrašyti grandinės minimizuotą funkciją pasinaudojant Karno diagramų analize.
2. Minimizuoti loginės funkcijos (duoda dėstytojas) lygtį pasinaudojant loginės algebros taisyklėmis ir, panaudojant gautą rezultatą, sudaryti loginę grandinę bei įprogramuoti ją į programuojamo loginio įrenginio (PLD) modulį.

Atliekant darbą reikia žinoti

Kas yra loginis elementas.

Įprastus loginių element tipus.

Loginės algebros tapatybes ir teoremas.

Kaip sudaromas Grejaus kodas.

Loginių funkcijų minimizavimo pasinaudojant Karno lentelėmis principą.

Nurodymai ir patarimai

Loginiai elementai

Skaitmeninėje elektronikoje visoms funkcijoms koduoti yra naudojamas dvejetainis kodas. Tokio kodo baziniai elementai yra 0 ir 1, kurie gali būti interpretuojami kaip loginės reikšmės *melas* (angl. *false*) ir *tiesa* (angl. *true*), atitinkamai. Elektrotechnikos atžvilgiu 0 ir 1 yra išreiškiami dviem skirtingais įtampos lygiais. Įprasta žemą įtampos lygį U^0 laikyti loginiu 0, o aukštą įtampos lygį U^1 – loginiu 1. Praktikoje gali pasitaikyti ir kitaip, priklauso nuo įtaiso dizaino.

Bazinis skaitmeninės elektronikos vienetas yra loginis elementas – tai elektroninis įtaisas, kuris turi vieną ar daugiau įėjimų, bent vieną išėjimą ir atlieka loginės algebros (angl. *Boolean algebra*) funkciją su įėjimo signalais, jo išėjime stebimas rezultatas. Tas faktas, kad loginėje algebroje galimos tik dvi argumento vertės, įgalina santykinai paprastai atlikti matematinės operacijas lyginant jas su įprastine algebra (nėra neigiamų skaičių, šaknų, logaritmų ir pan.). Iš tiesų dvejetainėje algebroje naudojamos trys pagrindinės loginės operacijos:

- 1) Loginė sudėtis – disjunkcija arba ARBA operacija ir žymima „+“ arba „v“ simboliais tarp argumentų;
- 2) Loginė daugyba – konjunkcija arba IR operacija, žymima „•“ „&“, „^“ simboliais tarp argumentų;
- 3) Loginis neigimas – inversija arba NE operacija, žymima „~“ simboliu virš kintamojo arba „~“ prieš jį.

Žemiau pateiktos 1 ir 2 kintamųjų loginių funkcijų reikšmių lentelės:

1 lentelė. Vieno kintamojo X loginių funkcijų $f_0...f_3$ reikšmių lentelė

| X | f_0 | f_1 | f_2 | f_3 |
|---|-------|-------|-------|-------|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |

2 lentelė. Dviejų kintamųjų X ir Y loginių funkcijų $f_0...f_{15}$ lentelė

| X | Y | f_0 | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 | f_9 | f_{10} | f_{11} | f_{12} | f_{13} | f_{14} | f_{15} |
|---|---|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|----------|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Analizuojant šias lenteles nesunku pastebėti, kad pusė visų loginių funkcijų yra antros pusės inversija, t.y. $f_0 = \neg f_3$ ir $f_1 = \neg f_2$ vienam kintamajam, $f_0 = \neg f_{15}$, $f_1 = \neg f_{14}, \dots, f_7 = \neg f_8$. Taigi, šios funkcijos yra ekvivalentiškos, todėl dažnai pakanka išnagrinėti tik vieną iš jų.

Vieno kintamojo funkcijos f_0 ir f_3 bei dviejų kintamųjų funkcijos f_0 ir f_{15} yra tiesiog konstantos 0 ir 1 atitinkamai.

Vieno kintamojo funkcija f_1 vadinama kintamojo inversijos funkcija, o $f_2 = X$ – jo kartojimo funkcija. Dviejų kintamųjų funkcijos $f_{12} = X$ ir $f_3 = \bar{X}$ vadinamos X kartojimu ir X inversija atitinkamai, analogiškai $f_{10} = Y$ ir $f_5 = \bar{Y}$ – Y kartojimu ir Y inversija.

Loginių funkcijų f_8 ir f_7 pora sudaro IR bei IR-NE funkcijas. Pastaroji dar vadinama Šeferio funkcija ir žymima Šeferio štrichu $X|Y$.

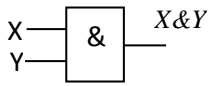
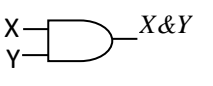
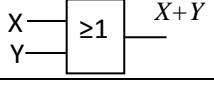
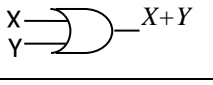
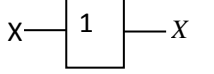
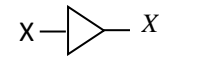
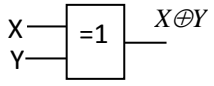
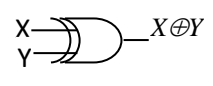
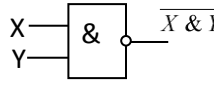
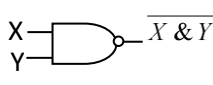
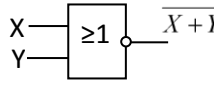
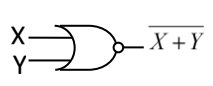
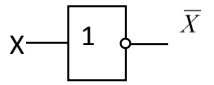
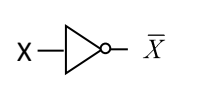
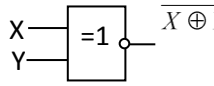
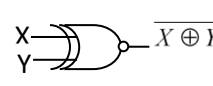
f_{14} – loginė ARBA funkcija, o f_1 – jos inversija, t.y. ARBA-NE funkcija, dar vadinama Pirsio funkcija ir žymima Pirsio strėle $X \downarrow Y$.

Dažnai taikoma loginė funkcija $f_6 = X \oplus Y$, vadinama išskirtinio ARBA (angl. *exclusive OR*) logine funkcija. Ji dar kartais vadinama logine sudėtimi vienoje skiltyje. Loginio ekvivalentiškumo funkcija $f_9 = \overline{X \oplus Y}$ yra išskirtinio ARBA inversija. Ji lygi vienetui tik tuomet, kai lygūs abu loginiai kintamieji.

Loginės funkcijos $f_2 = \bar{X}Y$ ir $f_4 = X\bar{Y}$ vadinamos X ir Y draudimu atitinkamai, o panašios funkcijos $f_{11} = \bar{X} + Y$ ir $f_{13} = X + \bar{Y}$ – implikacija nuo X prie Y ir nuo Y prie X atitinkamai.

Ne visos loginės funkcijos yra realizuotos gaminant jas mikroschemų pavidalu. Pramonėje gaminami loginiai elementai pateikti šioje lentelėje, t.y. baziniai loginiai elementai:

3 lentelė. Baziniai loginiai elementai

| Funkcija (2 lentelė) | Loginė išraiška | Loginio elemento pavadinimas | Loginio elemento žymėjimas schemoje | |
|-------------------------|-----------------------------|---------------------------------------|---|--|
| | | | Tarptautinis | Amerikietiškas |
| f8 | $f = X \& Y$ | IR (AND) |  |  |
| f14 | $f = X + Y$ | ARBA (OR) |  |  |
| f12 | $f = X$ | Kartotuvas (Buffer) |  |  |
| f6 | $f = X \oplus Y$ | Išskirtinis ARBA (XOR) |  |  |
| f7 | $f = \overline{X \& Y}$ | IR-NE (NAND) |  |  |
| f1 | $f = \overline{X + Y}$ | ARBA-NE (NOR) |  |  |
| f3 | $f = \overline{X}$ | Inverteris (Inverter) |  |  |
| f9 | $f = \overline{X \oplus Y}$ | Inversinis išskirtinis ARBA (XNOR) |  |  |

Skaitmeninėje elektronikoje loginiai elementai naudojami įtaiso valdymui sudaryti. Tam, kad užtikrinti norimą proceso eigą neretai tenka sudaryti logines grandines iš tam tikro skaičiaus įvairių loginių elementų. Sudėtinės loginės grandinės funkcija yra aprašoma lygtimi, pvz.: $f = XY + (\overline{X}Y \oplus X\overline{Y}Z)$. Skaičiuojant loginių išraiškų vertes reikia laikytis šių pagrindinių taisyklių: pirmiausia atliekamos atskirų loginių kintamųjų inversijos, po to veiksmai skliausteliuose, IR operacijos, ARBA operacijos, jei kelios operacijos yra invertuotos, pirmiausia atliekami atitinkami veiksmai, o po to gautas rezultatas invertuojamas. Sudarinėjant sudėtingas logines grandines neretai pasitaiko perteklinių elementų, kurie iš pirmo žvilgsnio nėra lengvai pastebimi. Norint išvengti perteklinio elementų skaičiaus loginėje grandinėje yra taikomi minimizavimo metodai. Du pagrindiniai yra minimizavimas naudojant loginės algebros tapatybes ir teoremas bei minimizavimas naudojant Karno (angl. *Karnaugh*) lenteles.

Loginės tapatybės ir teoremos

Loginės teoremos gali būti skirstomos į dvi grupes: vieno kintamojo ir keleto kintamųjų. Vieno kintamojo tapatybės ir teoremos yra:

$$\begin{array}{cccc}
 X \cdot 0 = 0 & X \cdot 1 = X & X \cdot X = X & X \cdot \bar{X} = 0 \\
 X + 0 = X & X + 1 = 1 & X + X = X & X + \bar{X} = 1
 \end{array}$$

Pirmos dvi daugybos ir sudėties teoremos yra trivialios. Trečiosios teoremos taip pat nereikalauja ypatingo aiškinimo, kadangi $1 \cdot 1 = 1$, $1 + 1 = 1$, o $0 \cdot 0 = 0$ ir $0 + 0 = 0$. Paskutiniosios taip pat gali būti įrodomos analogiškai: $1 \cdot 0 = 0$, $1 + 0 = 1$ ir $0 \cdot 1 = 0$ ir $0 + 1 = 1$.

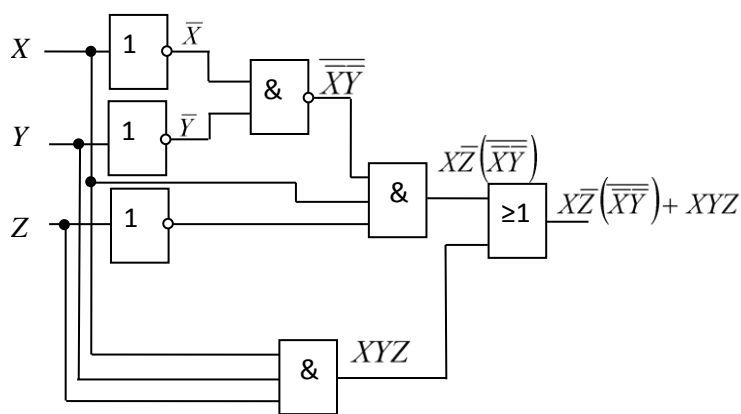
Keleto kintamųjų teoremos yra šios:

| | | |
|--|---|-----------------------|
| $X + Y = Y + X$ | $X \cdot Y = Y \cdot X$ | komutatyvumo dėsnis |
| $X + (Y + Z) = (X + Y) + Z = X + Y + Z$ | $X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z = X \cdot Y \cdot Z$ | asociatyvumo dėsnis |
| $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$ | $X + Y \cdot Z = (X + Y) \cdot (X + Z)$ | distributyvumo dėsnis |
| $X + X \cdot Y = X$ | $X + \bar{X} \cdot Y = X + Y$ | absorbavimo dėsnis |
| $\overline{X + Y} = \bar{X} \cdot \bar{Y}$ | $\overline{X \cdot Y} = \bar{X} + \bar{Y}$ | de Morgano teorema |

Komutatyvumo, asociatyvumo ir distributyvumo dėsniai gerai žinomi iš klasikinės algebros ir jų prasmė čia ta pati: komutatyvumo dėsnis skelbia, kad kintamųjų sudėties ar dauginimo tvarka nesvarbi kadangi rezultatas dėl to nepakinta, asociatyvumo dėsnis leidžia grupuoti dauginamuosius ar sumos narius norima tvarka, distributyvumas įgalina išplėsti išraiškas paeiliui dauginant jos narius. Kita naudinga distributyvumo savybė – galimybė iškelti bendrus daugiklius už skliaustų, pvz. $X\bar{Y}Z + \bar{Y}Z = \bar{Y}(XZ + Z) = \bar{Y}Z$. Absorbavimo dėsnis, panaudotas praeitame pavyzdyje, neturi analogų klasikinėje algebroje. Nepaisant to, tai naudingas dėsnis paprastinant įvairias išraiškas. Jo teisingumu galima įsitikinti pasinaudojant tuo pačiu distributyvumo dėsniu ir vieno kintamojo tapatybėmis: $X + XY = X(1 + Y) = X \cdot 1 = X$.

Bene svarbiausia loginės algebros teorema yra de Morgano dėsnis. Jis ypač naudingas prastinant išraiškas, kuriose yra dviejų ar daugiau kintamųjų sumos ar sandaugos inversija, t.y. šią taisyklę nesunku išplėsti keletui kintamųjų: $\overline{X + Y + Z + Q} = \bar{X} \cdot \bar{Y} \cdot \bar{Z} \cdot \bar{Q}$. Pavyzdys:

$$\overline{X\bar{Y} + Z} = \overline{X\bar{Y}} \cdot \bar{Z} = (\bar{X} + \bar{\bar{Y}}) \cdot \bar{Z} = (\bar{X} + Y) \cdot \bar{Z}$$



1 pav. Schemos paprastinimas naudojant de Morgano teorema

Supaprastinsime schemą, pateiktą 1 pav. Pirmiausia panaudojame de Morgano teorema reiškiniui skliaustuose:

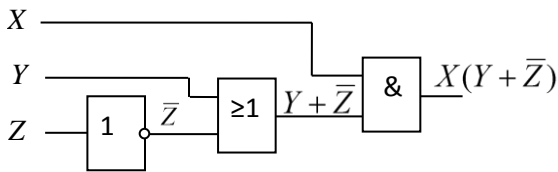
$$\overline{(\bar{X}\bar{Y})} = (\bar{\bar{X}} + \bar{\bar{Y}}) = (X + Y)$$

Panaudoję distributyvumo dėsnį gauname:

$$\begin{aligned}
 X\bar{Z}(X + Y) &= X\bar{Z}X + X\bar{Z}Y \\
 &= X\bar{Z} + XY\bar{Z}
 \end{aligned}$$

Grupodami panašius narius gauname:

$X\bar{Z} + XY\bar{Z} + XYZ = X\bar{Z} + XY(Z + \bar{Z}) = X(\bar{Z} + Y)$. Supaprastinta šio reiškinio schema pateikta 2 pav.



2 pav. Supaprastinta 1 pav. schema

Karno diagramos

Analizuojant sudėtingas logines funkcijas, t.y. kai kintamųjų skaičius ≥ 4 , yra patogesnis minimizavimo būdas, vadinamas Karno diagramų metodu. Sudarant Karno diagramas naudojamos loginių funkcijų užrašymu loginių sandaugų suma. Kiekvienas sumos narys vadinamas mintermu ir žymimas m , o jo vertė lygi 1 jei reikšmių lentelėje tos eilutės funkcijos vertė lygi 1 ir 0 jei funkcijos vertė lygi 0. Sumuojant atitinkamus mintermus galime gauti bendrą loginį reiškinį aprašančią funkciją, pvz.:

| X | Y | f |
|---|---|---------|
| 0 | 0 | $f_0=0$ |
| 0 | 1 | $f_1=1$ |
| 1 | 0 | $f_2=1$ |
| 1 | 1 | $f_3=1$ |

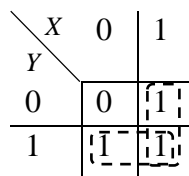
$F(X,Y)=f_0m_0+ f_1m_1+ f_2m_2+ f_3m_3$, kur $m_0 = \bar{X}\bar{Y}$, $m_1 = \bar{X}Y$, $m_2 = X\bar{Y}$ ir $m_3 = XY$.

Taigi $m_00+ m_11+ m_21+ m_31 = \bar{X}Y + X\bar{Y} + XY$.

Naudojantis algebriniu minimizavimo metodu gauname:

$\bar{X}Y + X\bar{Y} + XY = (\bar{X} + X)Y + X\bar{Y} = Y + X\bar{Y} = X + Y$.

Tą patį rezultatą galime gauti sukonstravę Karno diagramą. Tai kitokia reikšmių lentelės užrašymo forma, kurioje kiekvienas mintermas vaizduojamas kvadratėliu. Diagrama sudaroma taip, kad gretimuose kvadratėliuose skirtūsi tik vieno loginio kintamojo reikšmės – panaudojamas Grėjaus kodas. Grėjaus kodo sudarymas pateiktas 3 pav. Konstruojama įskaitant tai, kad kairysis ir dešinysis bei viršutinis ir apatinis diagramos kontūrai jungiasi tarpusavyje (naudojamos taip vadinamos periodinės kraštinės sąlygos). Taigi, aukščiau pateikto reiškinio Karno diagrama bus tokia:



Sudarius diagramą pažymimos visos vienetų poros, t.y. vienetai, esantys gretimuose stulpeliuose ar eilutėse (bet ne įstrižainėse). Tada loginė funkcija užrašoma apibrėžtų kintamųjų porų sandaugų suma atmetant tuos kintamuosius, kurių reikšmės poroje pakinta, t.y.:

$F(X,Y)=X+Y$.

Analizuojant sudėtingesnes išraiškas pažymimos gali būti ne tik poros, bet ir vienetai, ketvertai ir t.t., atmetant juose kintamuosius, keičiančius savo vertę. Kitas pavyzdys:

$$F(A, B, C) = \overline{A}\overline{B}\overline{C} + \overline{A}B + A\overline{B}\overline{C} + AC$$

| A | B | C | F |
|---|---|---|---------|
| 0 | 0 | 0 | $f_0=1$ |
| 0 | 0 | 1 | $f_1=0$ |
| 0 | 1 | 0 | $f_2=1$ |
| 0 | 1 | 1 | $f_3=1$ |
| 1 | 0 | 0 | $f_4=0$ |
| 1 | 0 | 1 | $f_5=1$ |
| 1 | 1 | 0 | $f_6=1$ |
| 1 | 1 | 1 | $f_7=1$ |

| AB \ C | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

Rezultatas: $F(A, B, C) = B + \overline{A}\overline{C} + AC$.

1. Rašoma 0, 1 stulpelis

2. Piešiamas veidrodėlis po stulpeliu

3. Užrašomas stulpelio atspindys

4. Prirašomi 0 prie skaičių virš veidrodėlio

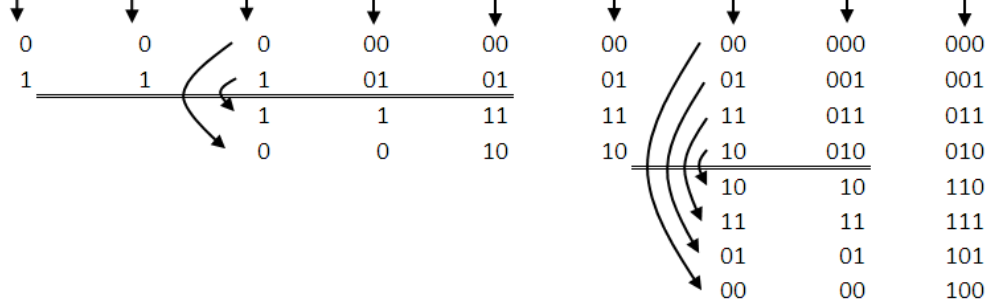
5. Prirašomi 1 prie skaičių po veidrodėliu

6. Gautas 2 bitų Grejaus kodas

7. Norint sudaryti 3 bitų Grejaus kodą piešiamas veidrodėlis po 2 bitų kodu

8. Virš veidrodėlio surašomi 0

9. Po veidrodėliu surašomi 1



3 pav. Grejaus kodo sudarymo schema

Darbo eiga

Darbui atlikti naudojamas 4 įėjimų ir 4 išėjimų loginės grandinės maketas prijungtas prie PLĮ modulio (4 pav.). Makete esantys žali šviestukai yra išėjimo indikatoriai, raudoni šviestukai – įėjimo indikatoriai. Šviečiantis šviestukas reiškia loginį 1, o nešviečiantis – loginį 0 atitinkamame įėjime ar išėjime. Kiekvienas įėjimas turi šliužiklinį jungiklį, kuriuo nustatoma įėjimo loginė būsena. Jungiklio padėtis arčiau šviestuko atitinka loginį 1, būsena perštūmus jungiklį toliau nuo šviestuko atitinka loginį 0.

Atliekant pirmąją darbo užduotį reikia sudaryti visas įmanomas įėjimo kombinacijas panaudojant maketo įėjimo jungiklius ir stebint kas vyksta išėjimuose, sudaryti loginės grandinės reikšmių lentelę. Tuomet, panaudojant gautus rezultatus, kiekvienam išėjimui sudaroma Karno

diagrama ir pasinaudojant ja užrašoma minimizuota kiekvieno išėjimo funkcija. Karno analizė atliekama laboratorinio darbo metu.

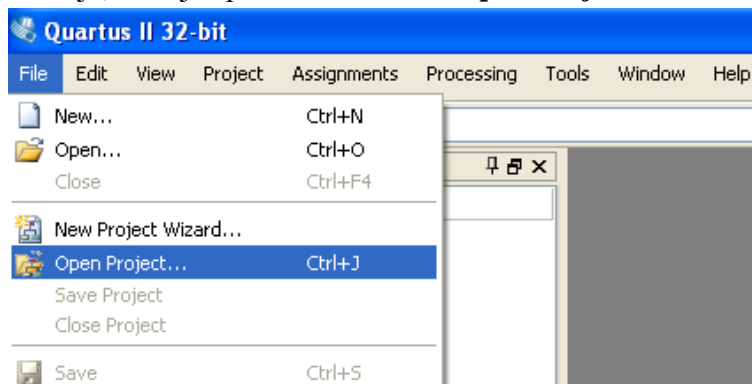
Gavus teisingą rezultatą pirmojoje užduotyje, dėstytojas duoda trijų kintamųjų loginę funkciją, reikalingą antrosios užduoties atlikimui. Gautą funkciją reikia minimizuoti. Rekomenduojama naudoti loginės algebros tapatybes ir teoremas. Teisingai minimizavus gautą funkciją, ja reikia įprogramuoti į PLĮ modulį. Darbas atliekamas kompiuteriu. PLĮ modulis prijungiamas prie kompiuterio per USB lizdą.

Nurodymai darbui kompiuteriu

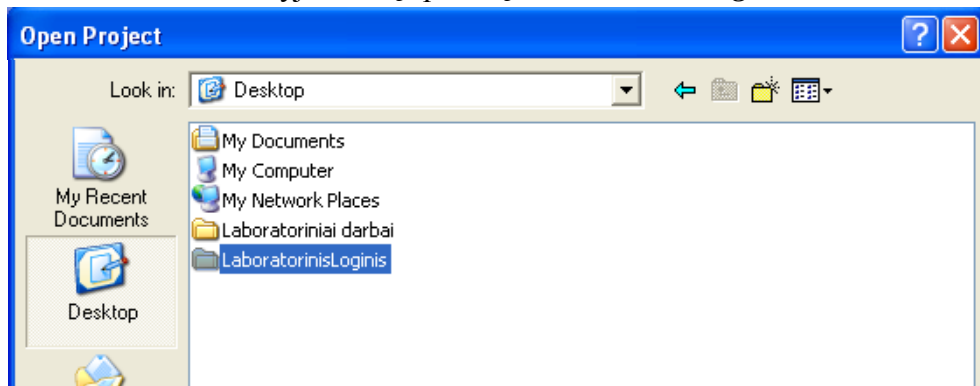
Darbalangyje surandame nuorodą į programą **Quartus II** ir įjungiamo PLĮ programavimo redaktorių.



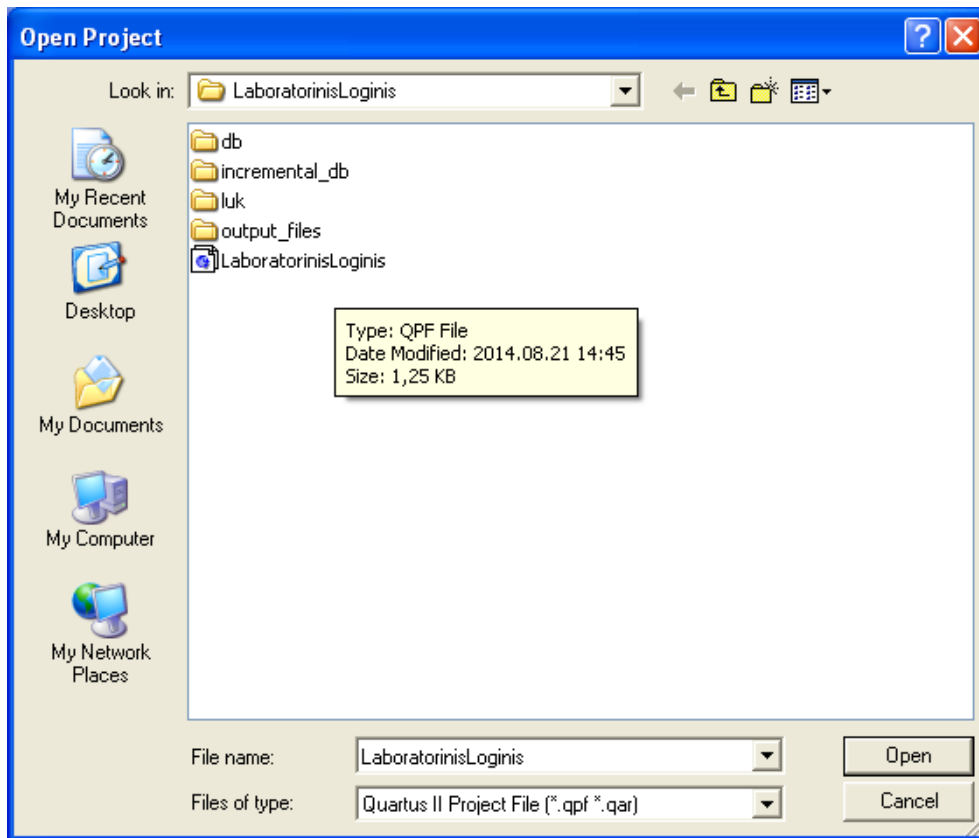
Viršuje, kairėje spaudžiame **File > Open Project**



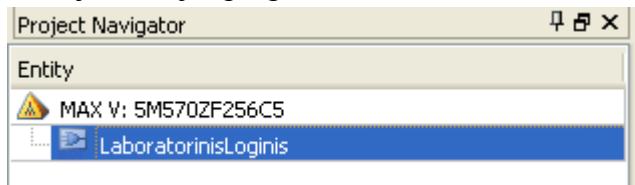
Atidarome darbalaukyje esantį aplanką **LaboratorinisLoginis**



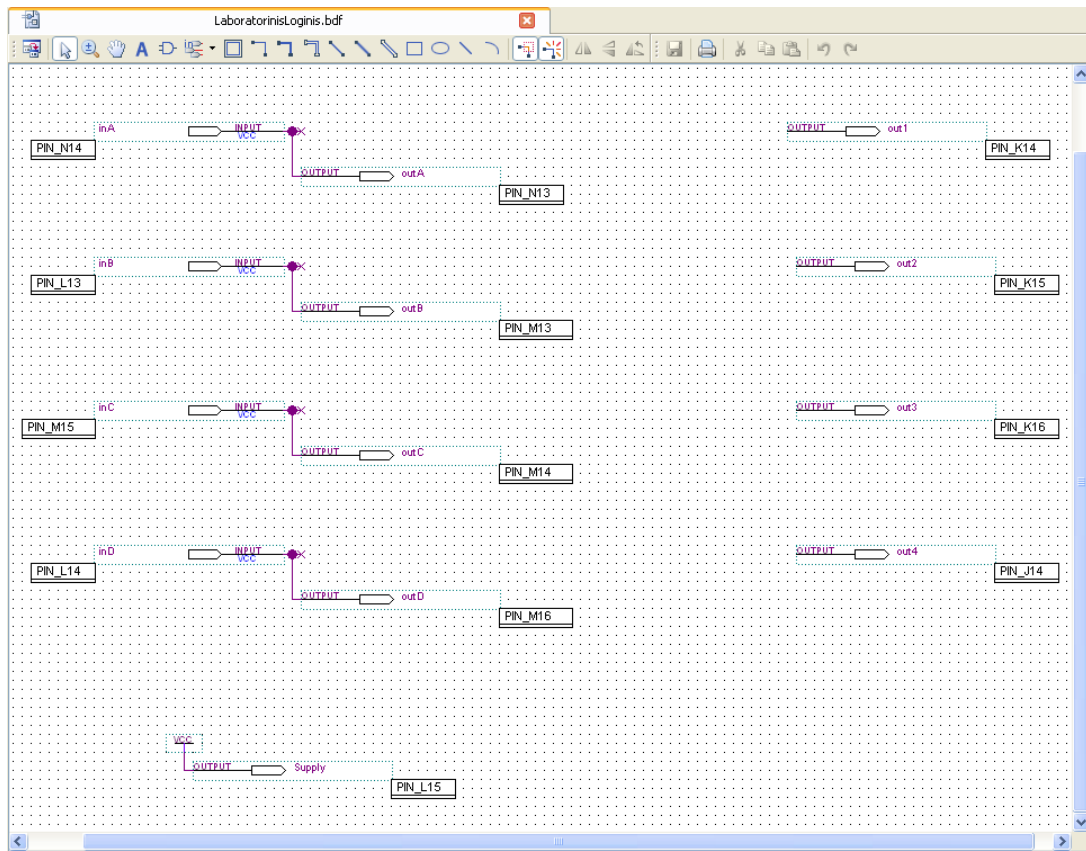
Pasirenkame laikmeną **LaboratorinisLoginis.qpf** ir spaudžiame **Open**.



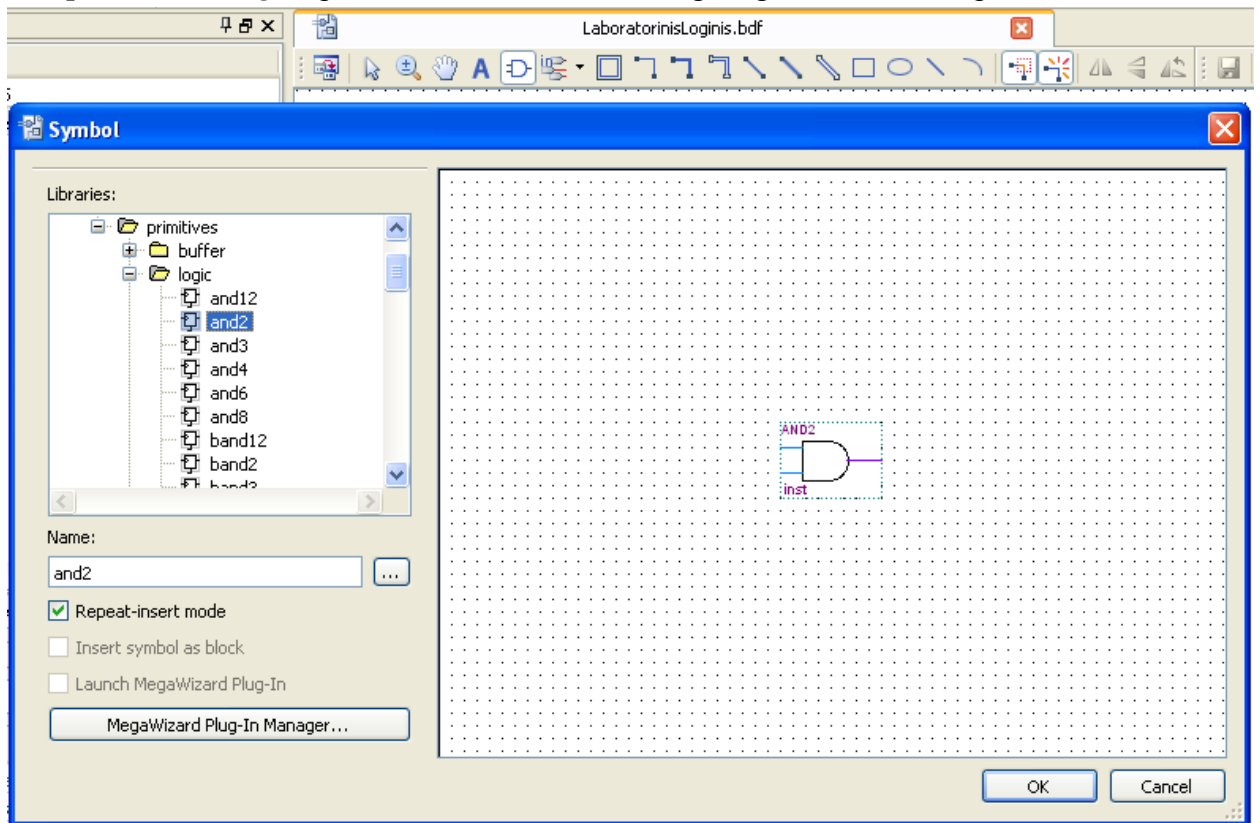
Kairėje, viršuje spragtelime du kartus ant laikmenos *LaboratorinisLoginis.bdf*



Atsiveria loginių grandinių redaktorius. Galima pastebėti, kad jau yra sudaryti įėjimų ir išėjimų kontaktai. Jei įėjimo arba išėjimo kontaktai redaktoriuje yra ištrinami ar kitaip sugadinami, reikia išjungti **Quartus II**, gavus užklausą „ar norite išsaugoti?“, spausti **No** ir įjungti programą bei atsidaryti projektą iš naujo.



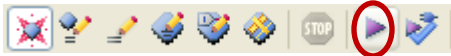
Pasirenkame **Symbol Tool**. Atsiveria langas, kuriame galima rasti reikalingus loginius elementus *primitives* > *logic* aplanke. Pasirenkame reikalingą loginį elementą ir spaudžiame **OK**.



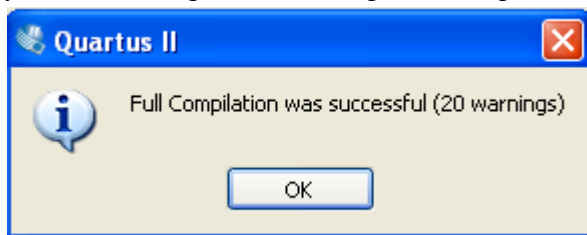
Kairiuoju pelės klavišu loginis elementas pastatomas norimoje grandinės vietoje. Sudėjus pakankamą kiekį elementų, klavišo *esc* paspaudimu grąžiname kursoriaus būseną į neutralią. Norint grandinės elementą pašalinti, jį pažymime ir spaudžiame *delete* klavišą.

Grandinės elementai sujungiami linijomis. Kursorius nuvestas prie kontakto taško pasikeičia į „+“, tuomet spaudžiame kairįkį pelės klavišą ir jo neatleisdami tempiame link kito kontakto taško arba bet kokio taško, kuriame norime, kad liktų atviras kontaktas. Naudojame tik reikalingą įėjimų ir išėjimų skaičių.

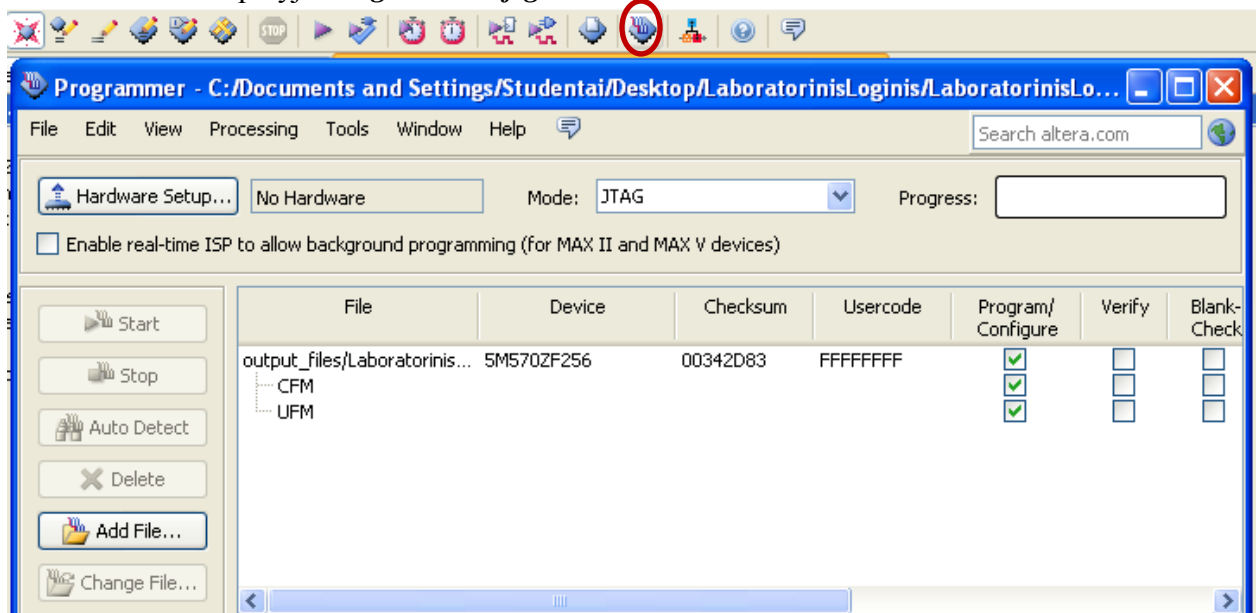
Sujungę grandinę spaudžiame **Start Compilation**.



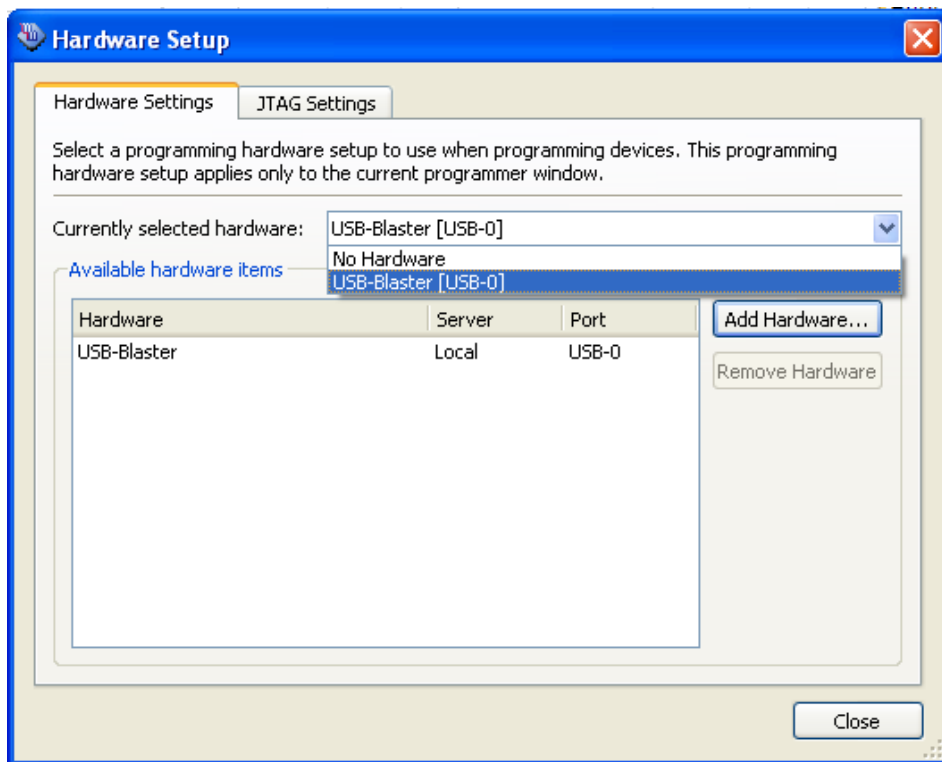
Kompiuterio ekrane pasirodžius išsaugojimo užklausa spaudžiame **Yes**. Jei grandinė sudaryta teisingai, turi pasirodyti apie sėkmę pranešanti lentelė. Kitu atveju reikia ištaisyti padarytas klaidas ir pakartoti kompiliavimo procedūrą.



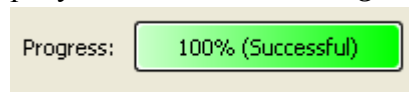
Sėkmingai sukompiliavus spaudžiame **Programmer**, atsiveria programatoriaus langas. Uždedame varneles stulpelyje **Program/Configure**



Jei programatoriaus langas rašo **No Hardware**, patikriname ar PLĮ modulis prijungtas prie kompiuterio USB kabeliu. Tuomet spaudžiame **Hardware Setup**. Atsidariusiame lange, laukelyje **Currently selected hardware** parenkame **USB-Blaster** spaudžiame **Close**.



Programatoriaus lange spaudžiame **Start**. Vyksta PLĮ programavimas. Jei programavimas pavyko sklandžiai, **Progress** laukelyje turi pasirodyti užrašas **100% (Successful)**.



Užprogramuoto modulio veikimą reikia patikrinti makete ir sudaryti loginės grandinės reikšmių lentelę bei nusipiešti sudarytą loginę grandinę.

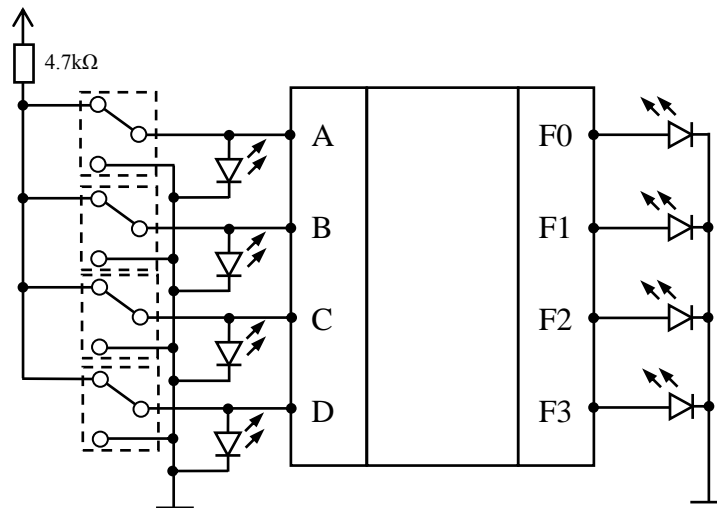
Darbo ataskaitoje pateikti

- Keturių įėjimų ir keturių išėjimų loginės grandinės reikšmių lentelę ir atliekamos loginės funkcijos minimizuotas lygtis bei tarpinius skaičiavimo rezultatus.
- Trijų kintamųjų loginės funkcijos minimizuotą variantą ir rezultato išvedimą.
- Minimizuotos funkcijos loginę grandinę.
- Rezultatų aptarimą ir išvadas.

Literatūra

1. <http://www.allaboutcircuits.com/> > Digital > Chapter7: Boolean Algebra
2. <http://www.allaboutcircuits.com/> > Digital > Chapter8:Karnaugh Mapping
3. Pramoninė elektronika. T. 2. Impulsinės bei skaitmeninės schemas ir energetiniai bei technologiniai įtaisai / A. Lašas, B. Bartkevičius, P. Šurna ir kiti. - V.: Mokslas, 1991. - 254 p.
4. Loginės schemas / R. Kirvaitis – V.:Enciklopedija, 1999. – 255 p.

Priedas



4 pav. Darbo maketo schema

Maketo išvadų paskirtis

A...D – loginiai įėjimai

F0...F3 – loginiai išėjimai