

ALGIRDAS KLIMAVIČIUS

PLM *ALTERA*

MOKOMASIS METODINIS LEIDINYS

**VIKO ELEKTRONIKOS IR INFORMATIKOS FAKULTETAS
2008**

TURINYS:

1. PLM APŽVALGA	5
2. PLM SCHEMOTECHNINĖ DALIS	9
2.1 Paprasčiausių programuojamų loginių įtaisų sandara	9
2.2 Programuojamos loginės matricos tipinės funkcinės schemos analizė, veikimo principo, pakopų ir elementų paskirties paaiškinimas	13
2.3 Programavimo proceso analizė ir principo paaiškinimas	18
3. PLM KONSTRUKCIJA	19
3.1 ALTERA MAX3000 serijos PLM konstrukcijų apžvalga	19
3.2 ByteBlaster MV konstrukcija	21
4. PROJEKTAVIMO SISTEMA MAX+plus II	24
4.1 Pagrindiniai projekto sukūrimo principai	24
4.2 Redaktorių paaiškinimas	24
4.3 Darbas su projektu	29
5. PRAKTINIAI PLM PROGRAMAVIMO DARBAI	32
5.1 Darbas su AHDL programine kalba	32
5.2 Darbas su VHDL programine kalba	34
5.3 Tipinių kombinacinių grandinių modeliavimas	37
5.3.1 Šifratoriaus modeliavimas tekstiniu būdu	37
5.3.2 Šifratoriaus modeliavimas grafiniu būdu	43
5.3.3 Dešifratoriaus modeliavimas tekstiniu būdu	44
5.3.4 Dešifratoriaus modeliavimas grafiniu būdu	45
5.3.5 Multiplekserio modeliavimas tekstiniu būdu	47
5.3.6 Multiplekserio modeliavimas grafiniu būdu	51
5.3.7 Demultiplekserio modeliavimas tekstiniu būdu	52
5.3.8 Demultiplekserio modeliavimas grafiniu būdu	55
5.4 Skaitiklio modeliavimas	56
5.4.1 Skaitiklio modeliavimas tekstiniu būdu	56
5.4.2 Skaitiklio modeliavimas grafiniu būdu	61
6. ALTERA INTERNETE	65
7. INFORMACIJA APIE ALTERA PRODUKTUS INTERNETE	66
Purchasing Information	66
Cyclone II FPGA Starter Development Kit	68
Ordering Information	68

Development Kit Contents	68
Available Documentation.....	69
Cyclone III FPGA Starter Kit from <i>Altera Corporation</i>	70
Ordering Information	70
Cyclone III and Nios II 32-bit soft IP processor	71
High-Speed Mezzanine Card Interface	71
Development Kit Contents	71
Available Documentation.....	72
DSP Development Kit, Cyclone II Edition	72
Ordering Information	72
Development Kit Contents	73
Available Documentation.....	74
MAX II Development Kit from <i>Altera Corporation</i>	74
Ordering Information	74
Development Kit Contents	74
Nios II Development Kit, Cyclone II Edition	75
Development Kit Contents	76
Software Development Tools.....	77
PCI Development Kit, Cyclone II Edition	Error! Bookmark not defined.
Ordering Information	78
Development Kit Contents	78
Software Development Tools.....	79
Hardware Development Tools	79

1. PLM APŽVALGA

Visų gaminamų PLM bendros savybės yra tai, kad jie visi apdorojant signalus:

- dirba labai greitai
- yra galimybė realizuoti sudėtingus paralelinius algoritmus
- yra automatizuoto projektavimo sistemos, su kuriomis galima atlikti pilną sistemos modeliavimą
- yra galimybė programuoti arba keisti konfigūraciją tiesiogiai sistemoje
- yra palaikymas keičiant algoritmus į kalbą VHDL, AHDL lygio
- yra palaikymas pagal lygius ir galimybė realizuoti standartinę sąsają
- yra galimybė iškart panaudoti funkcijas, kurių algoritmai yra labai sudėtingi
- PLM architektūrinės savybės gali būti labai gerai pritaikytos tokioms operacijoms kaip daugyba, kodavimas

Dabar PLM darbo greitis jau pasiekė tokią ribą, kaip 250-300MHz, kas leidžia realizuoti daugumą algoritmų radijo diapazone.

Tam, kad geriau suprastume PLM skirtumus, reikia apžvelgti jų vystymąsi. 1970 metais integralinių schemų rinkoje dominavo schemas, kurios turėjo programuojamas matricas „IR“ ir „ARBA“. Šios architektūros pavadinimas buvo FPLA (Field Programmable Logic Array) ir FPLS (Field Programmable Logic Sequencers). Šių PLM analogai buvo K556PT1, PT2, PT21. Šios architektūros trūkumas – tai mažas „ARBA“ dalies panaudojimas.

Tobulėjant šiai struktūrai integralinių schemų kūrėjai pasiūlė paprastesnio tipo architektūros sprendimą – programuojamos loginės matricos architektūrą (PAL – Programmable Array Logic ir GAL – Gate Array Logic) – tai PLM, kurie turi programuojamą „IR“ matricą ir fiksuotą „ARBA“ matricą. PLM GAL tipo išėjime yra trigeris. Šioje klasėje yra daug PLM su nedideliu integracijos lygiu. Pavyzdžiui, tai gali būti ИС KM155XII4, XII6, XII8, XII8 arba pradiniai gaminiai 1980 metais tokių bendrovių kaip INTEL, ALTERA, AMD, LATTICE ir kitų.

Taip pat buvo sukurtas PLM architektūros tipas, kuris turėjo tik programuojamą matricą „IR“. Tokios matricos pavyzdys gali būti INTEL bendrovės 85C508 schema.

Kitas sprendimas sumažinti programuojamą matricą „ARBA“ yra programuojama makrologika. PLM, sukurti pagal šią architektūrą, turi vienintelę programuojamą matricą „IR-NE“ arba „ARBA-NE“. Naudojant daug inversinių atbulinių ryšių, yra galimybė ir šiose PLM formuoti sudėtingas logines funkcijas. Tokį tipą turi PLM PLHS501 ir PLHS502 schemų SIGNETICS bendrovės, turintiems „IR-NE“ matricos tipą. „ARBA-NE“ matricos tipo pagrindas yra bendrovės EXEL schema XL78C800.

Visos anksčiau išvardintos PLM architektūros turi nedidelį narvelių kiekį ir naudojamos realizuojant nesudėtingus įtaisus, kuriems nėra integralinių schemų analogų. Aišku, norint realizuoti sudėtingus valdymo algoritmus, jų panaudojimas nėra įmanomas.

1980m. PLM rinkoje dominuoja trys bendrovės. 1983 buvo įsteigta ALTERA Corporation bendrovė, (101 Innovation Drive, San Jose, CA 95134, USA, www.ALTERA.com). 1984m. įsteigta Xilinx Inc. bendrovė (2100 Logic Drive, San Jose, CA 95124-3400, USA, www.xilinx.com). 1985m. įsteigta Actel Corporation bendrovė (955 East Arques Avenue, Sunnyvale, CA 94086-4533, USA, www.actel.com). Šios trys bendrovės valdo iki 80% visos PLM rinkos. Jeigu anksčiau PLM buvo vienu iš daugelio produktų, kuriuos gamino tokios kompanijos kaip INTEL, AMD ir kiti, tai pradedant nuo 1980 -1990 metų PLM rinkoje vykdoma specializacija ir pagrindinius parametrus jau užduoda tos bendrovės, kurios užsiima tik PLM kūrimu ir gaminimu.

Atsiradus naujiems gamintojams, atsirado ir naujos architektūros. Integralinės schemos PLM turi tokį architektūros tipą, koks labai patogus realizuojant skaitmeninius aparatus. Šis architektūros vystymas – CPLD (Complex Programmable Logic Device). Tai PLM, kuri turi keliolika loginių blokų (LB), kurie sujungti panaudojant komutacinę matricą. Kiekvienas loginis blokas atvaizduojamas kaip atskira PLM, šiame bloke yra programuojama matrica „IR“ ir fiksuota matrica „ARBA“. PLM CPLD tipas dažniausiai turi didelį integracijos lygį (iki 10000 ekvivalentinių ventilių, iki 256 makronarvelių). Šiai integralinių schemų klasei priklauso ALTERA bendrovės schemos PLM MAX5000 ir MAX7000, XILINX bendrovės schemos XC7000 ir XC9500 ir taip pat daug kitų mikrochemų gamintojų (Atmel, Vantis, Lucent).

Didelis vystymasis šioje rinkoje davė savus rezultatus ir dabar panašios paskirties PLM gamina ir Xilinx bendrovė. Pačių geriausių (High End) šios bendrovės PLM yra Virtex klasės. Virtex klasė charakterizuojama plataus pasirinkimo greitu trasavimo resursu, yra atskiras operatyviosios atminties blokas, gera pagreitinamo pernešimo logika. Šios serijos PLM palaiko didelį darbo dažnį – iki 200MHz (standartas HSTL IV). Serijos Virtex gaminiai dėl geros gamavimo technologijos ir gero verifikacijos proceso turi sąlyginai nedidelę kainą (iki 40% ekvivalentinės serijos XC4000XL kainos). Taip pat Xilinx bendrovė gamina XC3000A, XC4000E, Spartan, XC5200, CPLD XC9500 ir CoolPLD serijos PLM, kurie vartoja mažai energijos. Bendrovė duoda galimybę nemokamai naudotis WebPACK programiniu paketu, kuris palaiko CPLD XC9500 ir CoolPLD serijos PLM. Jis leidžia naudojant VHDL kalbą programuoti PLM. Kitų PLM programavimas įmanomas naudojant kitų versijų programinę įrangą (kūrė Synopsys bendrovė).

Kita bendrovė, kuri gamina panašios paskirties PLM, yra Actel Corporation. Dabar Actel bendrovės PLM ypatumas yra tas, kad juose naudojama Antifuse technologija – PLM turi metalizuotą trumpiklį, kuris naudojamas programavimo metu. Ši technologija duoda didelį patikimumą ir nereikalingas konfigūracinis pastovios atminties įtaisas. Pagal šią technologiją

gaminami ACT1, ACT2, 1200XL serijos PLM. Taip pat gaminamos ir naujos serijos – 54SX, A40MX ir A42MX (su integruota atmintimi). Šios schemos turi gerus rodiklius kaina/loginė talpa (PLM kuri pakeičia 300 – 350 TTL korpusu kainuoja 10\$, su 250MHz darbo dažniu).

Nauja Actel bendrovės ProASIC serija turi iki 500000 ekvivalentinių ventilių talpą. Šių įtaisų ypatumai – priklausomybė nuo maitinimo dėl FLASH technologijos panaudojimo ir integruoto atminties įrenginio egzistavimas. Projektuojant šias ir kitas PLM naudojamas programinis paketas Actel DeskTOP, kuriame yra sistemos modeliavimo galimybė, teksto VeriBest generavimo galimybė ir sintezė su Synplicity priedu. Actel bendrovės mikroschemos, kurios gaminamos pagal Antifuse technologiją, reikalauja specialaus tipo programatoriaus. Bet šios mikroschemos turi didelį patikimumą, todėl jos yra perspektyvios specialiam panaudojimui. Pavyzdžiui, PLM RH1280 serijos turi tokias charakteristikas:

- galima spinduliavimo dozė 300000 RAD
- loginė talpa – 16000 ekvivalentinių ventilių
- darbo greitis iki 135MHz buvo panaudotas Marse Tathindes sistemoje, kuri valdė ir apdorojo informaciją, kuri buvo iš skaitmeninės kameros. Taip pat ši sistema formavo kadra ir siuntė jį į Žemę.

Nagrinėsiu Alteros bendrovės PLM. Dabar ši bendrovė gamina PLM MAX3000, MAX7000, MAX9000 serijos. Papildomas faktorius renkantis Alteros bendrovės PLM yra tas, kad ši bendrovė siūlo gerus nemokamus programinius paketus. Daugumą programinių paketų galima nemokamai nusikopijuoti iš www.altera.com puslapio arba gauti su kompaktiniu disku CD ALTERA Digital Library, kuriame yra pilnas dokumentacijos rinkinys apie architektūrą ir jos panaudojimą PLM.

Taip pat Alteros bendrovės PLM išleidžiami su galimybe programuoti juos tiesiogiai ant plokštės. Programavimui ir įtaisų konfigūracijos pakrovimui naudojamas ByteBlaster ir ByteBlasterMV laidas. Reikia pabrėžti, kad naujos konfigūracinės pastovios atminties įtaisai EPC2 leidžia programuoti su šio įtaiso pagalba. Dėl to fakto programatorius tampa nereikalingas. Tas faktas sumažino viso technologinio proceso kainą.

Diplominiame darbe bus apžvelgtos PLM MAX3000 serijos. Jos architektūra yra panaši į MAX7000 serijos PLM, bet yra tam tikri skirtumai. Mikroschemos MAX3000 serijos pagaminti pagal CMOS EPROM technologiją su technologinėmis normomis 0.35μm, kas leido sumažinti jų kainą lyginant su MAX7000S serijos PLM. Visi MAX3000 serijos PLM palaiko sistemos (ISP, In-system programmability) programavimo technologiją ir periferinį skanavimą (boundary scan) pagal IEEE std. 1149.1 JTAG standartą.

1. Lentelė. Bendri PLM MAX3000 parametrai

	EPM3032A	EPM3064A	EPM3128A	EPM3256A
Ekvivalentinių ventilių loginė talpa	600	1250	2500	5000
Makronarvelių kiekis	32	64	128	256
Loginių blokų kiekis	2	4	8	16
Krogramuojamų išvadų kiekis	34	66	96	158
Signalų įėjimas-išėjimas sklidimo užlaikymas, t_{PD} , ns	4.5	4.5	5	6
Bendro taktinio signalo nustatymo laikas, t_{SU} , ns	3.0	3.0	3.2	3.7
Bendro taktinio signalo užlaikymas iki išėjimo, t_{CO1} , ns	2.8	2.8	3.0	3.3
Maksimalus bendras taktinis dažnis, f_{CNT} , MHz	192.3	192.3	181.8	156.3

Detaliam nagrinėjimui bus parinkta programuojama loginė EPM3032A tipo matrica, nes ji yra viena iš paprasčiausių MAX3000 serijos ir jos charakteristikos yra priimtinos skaitmeninių grandinių laboratorijoje.

MAX3000 serijos PLM įėjimo-išėjimo elementai leidžia dirbti sistemose su 5V, 3.3V, 2.5V signalų lygiu. Matrica turi tolydų struktūros tipą, kas leidžia realizuoti signalo sklidimo užlaikymo laiką iki 4.5 ns. PLM MAX3000 turi galimybę išėjimų aparatinei simuliacijai su atviru kolektoriumi (open – drains pin) ir palaiko PCI standartą. Yra galimybė pasirinktinai programuoti numetimo grandines, nustatinėti ir taktuoti trigerius, kurie įeina į makronarvelį. Galima panaudoti sumažinto energijos vartojimo režimą. Programuojamas loginis plėstuvus leidžia realizuoti viename makronarvelyje iki 32 funkcijos kintamųjų. Yra galimybė užduoti slaptumo bitą (security bit) norint apsaugoti informaciją nuo nesankcionuoto tiražo.

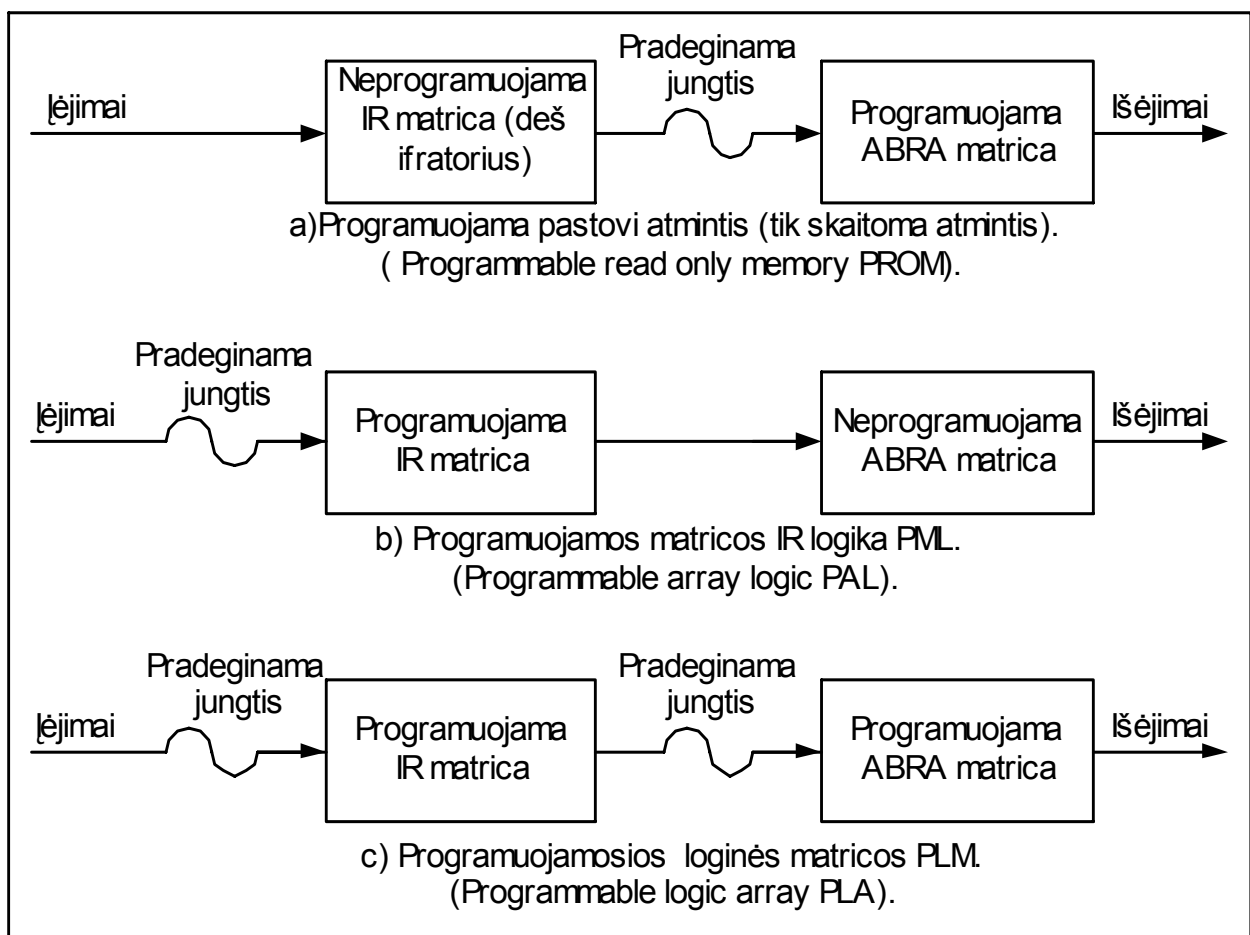
2. PLM SCHEMOTECHNINĖ DALIS

2.1 Paprasčiausių programuojamų loginių įtaisų sandara

Šioje dalyje apžvelgsime paprasčiausių programuojamų loginių įtaisų sandarą

PLĮ sudaryti iš IR ir ARBA loginių matricių – loginėmis matricėmis vadiname kelis ar keliolika vienodų loginių elementų, kurių atitinkami įėjimai sujungti (t.y. tarpusavyje sujungti visi pirmi įėjimai, visi antri it t.t.). IR ir ARBA įėjimuose yra lengvai sunaikinamos elektrinės jungtys t.y. lydūs saugikliai. Įtaisai programuojami nutraukiant programavimui skirtus elektrinius ryšius. Programuojamos jungtys, veikiant pakankamai didelei išorinei įtampai, pradeginamos (sudeginami saugikliai). Programuoti galima tik vieną kartą, nes nutrauktų ryšių atstatyti nebeįmanoma.

Yra trys PLĮ tipai, jie skiriasi pagal saugiklių išdėstymą IR ir ARBA matricose. Trys programuojamų loginių įtaisų tipai ir saugiklių vieta juose parodyta 2.1 pav.



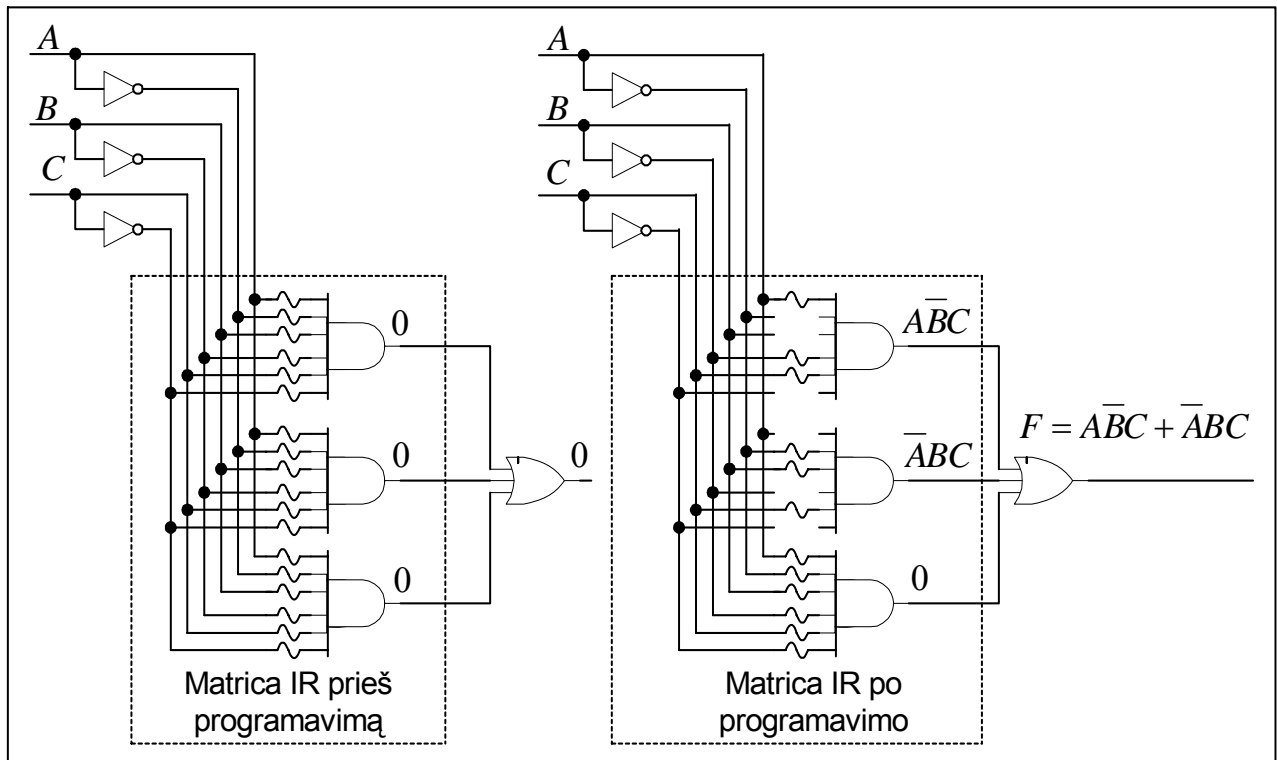
2.1 pav. Programuojamų loginių įtaisų rūšys

Dažniausiai iš šių matricių tipų naudojame vienos programuojamos matricos IR logiką PML, kuri sudaryta iš programuojamos IR matricos ir neprogramuojamos ARBA matricos; ir programuojamąsias logines matricas, kur programuojamos abi matricos.

Abiejų minėtų programuojamų įtaisų įėjimuose yra įėjimo grandys, kurios perduoda tiesioginius ir invertuotus įėjimų signalus, o įtaisų išėjimuose dažniausiai yra programuojamos išėjimo grandys, kurios gali išduoti tiesioginį arba invertuotą signalą.

2.2 pav. matome labai paprastą trijų įėjimų programuojamą loginį įtaisą PML, kuris sudarytas iš:

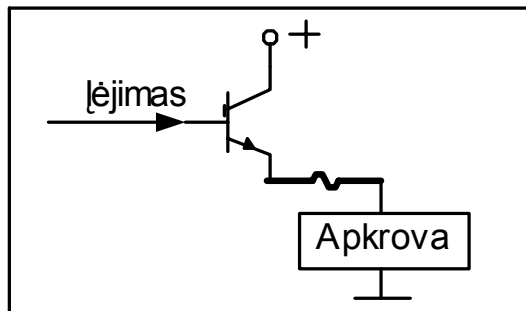
1. įėjimo grandies - 3 inverteriai;
2. programuojamos matricos IR, iš trijų, šešių įėjimų IR elementų;
3. elemento ARBA.



2.2 pav. PML prieš programavimą ir po programavimo

Loginių elementų IR įėjimuose yra lydūs saugikliai, kuriuos programavimo metu nutraukus, išėjime galime gauti trijų loginių sandaugų loginę sumą. Kiekviena sandauga, priklausomai nuo programavimo, bus kombinacija, sudaryta iš trijų kintamųjų - ABC, kurių kiekvienas gali būti tiesioginėje ar inversinėje formoje. 2.2 pav. matome, kaip atrodo grandinė prieš programavimą ir po programavimo. Nagrinėdami SIG serijas visas grandines sudarinėjome iš serijos bazinių elementų, PLĮ naudojami kiti scheminiai sprendimai, todėl čia netinka tvirtinimas, kad neprijungtas loginių elementų IR, ARBA įėjimas, atitinka loginio 1 prijungimą prie įėjimo. PLĮ. Šiose schemose programavimo metu nereikalingas įėjimas nepaliekamas neprijungtas, o tiesiog panaikinamas. (Įėjimas, kurio nėra, nepriima informacijos ir savaime aišku netrukdo).

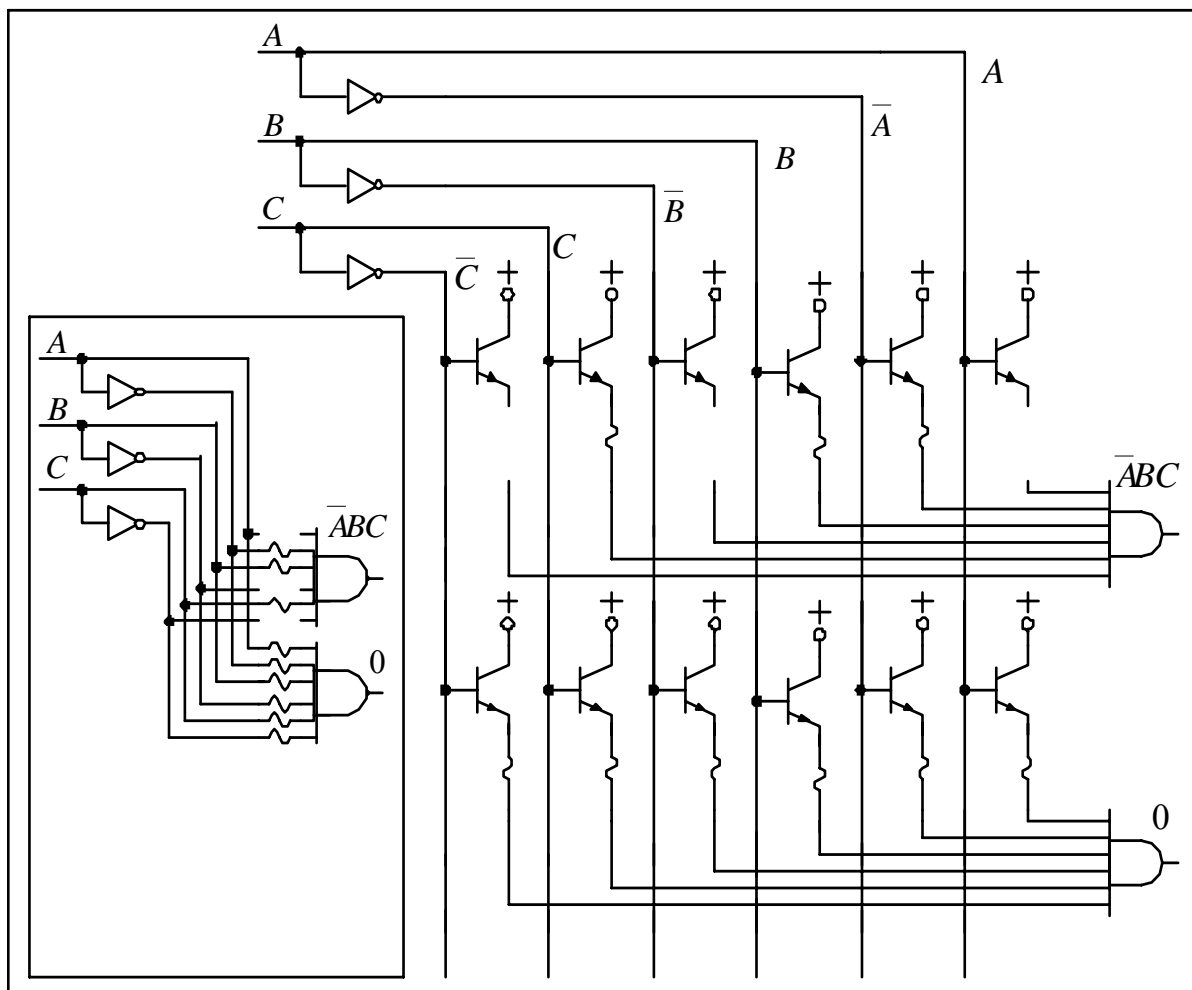
Dabar apie tirpiąsias jungtis. Joms galime panaudoti įvairius elektroninius įtaisus. 2.3 pav. matome emiterinį kartotuvą, kurio emiteris su apkrova sujungtas per nikelio – chromo lydinio saugiklį, kuris išsilydo, jei per jį tekanti srovė viršija leistiną, o tai ir vykdoma programavimo metu.



2.3 pav. Programuojamas elementas - emiterinis kartotuvas

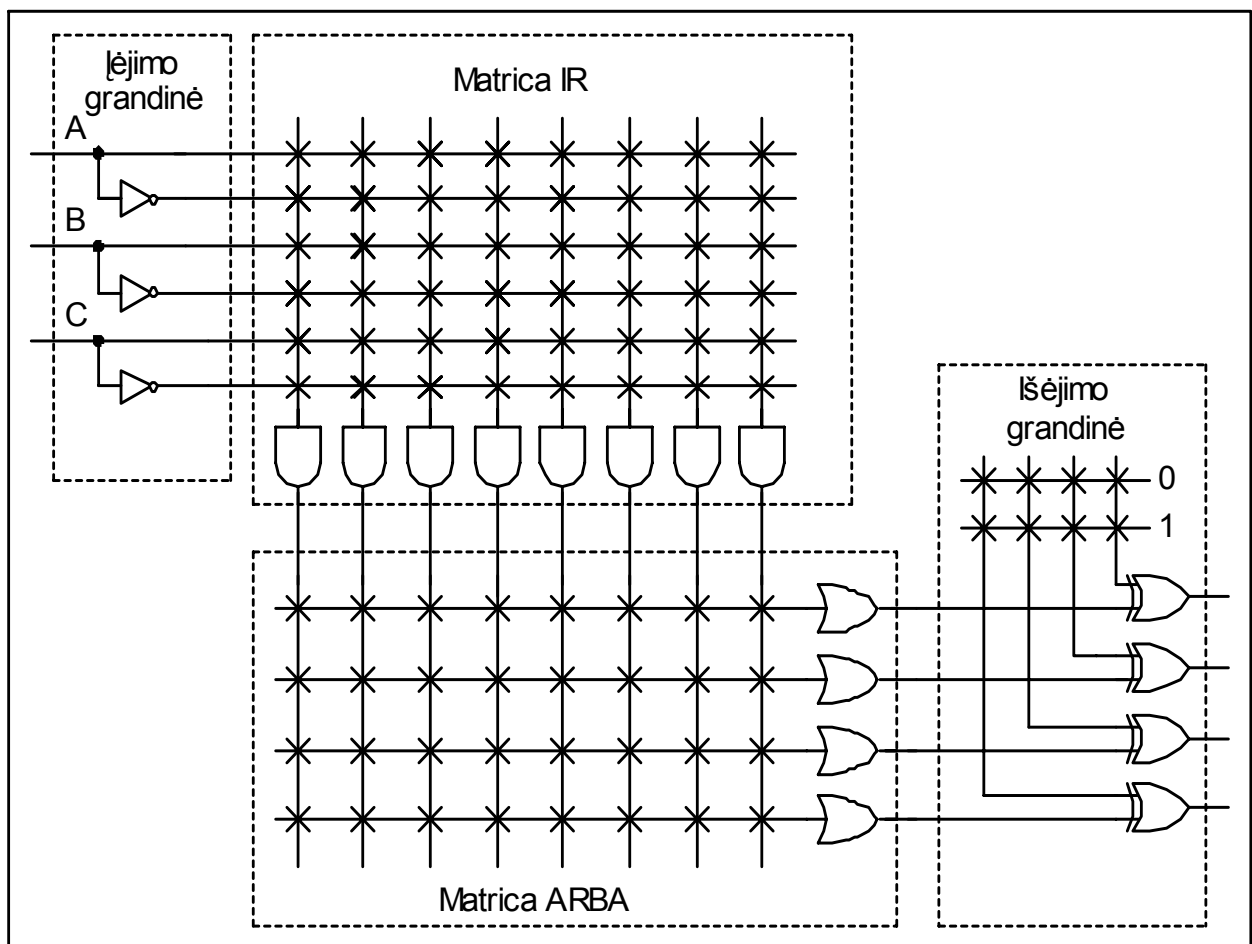
Toks programavimo metodas ir panaudotas 5.4 pav. pavaizduotoje schemoje. Palyginę šį paveikslėlį su 5.2 pav., matome, kad schema, papildyta emiteriniais kartotuvais, kurių įėjimai sujungti su įtaiso įėjimo grandimis, o išėjimai per saugiklius su atitinkamu IR įėjimu.

2.2 pav. ir 2.4 pav. pavaizduotos schemos PLĮ vaizdavimui nenaudojamos, toks vaizdavimo būdas buvo parinktas tik tam, kad geriau suprasti PLĮ vidinę sandarą.



2.4 pav. PML programuojama nutraukiant jungtis tranzistorių emiteriuose.

Kaip PLĮ vaizduojami realiose schemose matome 2.5 pav. Paveikslėlyje pavaizduota schema prieš programavimą, joje visi programuojami taškai pavaizduoti ženklu - ×. Jei programavimo metu jungtis nutraukiama, tai laidų susikirtimo vietoje joks ženkliukas nepiešiamas. Dažniausiai vaizduojant schemą prieš programavimą ženkliukai taip pat nepiešiami. Nes ne taip jau svarbu, kaip atrodo schema prieš programavimą – svarbiausia žinoti kaip programuoti ir kas gausis po programavimo. Be to realios schemos daug sudėtingesnės negu nagrinėjamos ir jų vidinė sandara rodoma tik žinynuose, o elektros – principinės schemose vaizduojami tik PLĮ žymenys, kaip ir visų kitų elementų.



2.5 pav. Trijų įėjimų PLM, kuri gali atlikti 8 logines sandaugas ir turi 4 išėjimus

2.5 pav. matome Programuojamųjų loginių matricų PLM loginę schemą, kuri sudaryta iš:

1. įėjimo grandinės trims kintamiesiems A, B, C;
2. programuojamos 8 loginių elementų IR matricos;
3. programuojamos 4 loginių elementų ARBA matricos;
4. išėjimo grandinės.

PLM įėjimo grandinė turi 3 įėjimus kintamiesiems A, B ir C ir 6 išėjimus tiesioginiams ir invertuotiems įėjimo signalams.

Visi 8 matricos IR loginiai elementai turi po 6 įėjimus (t.y. tiek yra įėjimo grandinės išėjimo signalų), kurie sujungti su tiesioginiais ir invertuotais įėjimo signalais. Visi 6 IR įėjimai pavaizduoti kaip vienas laidas su 6 programuojamais taškais.

Visi 4 matricos ARBA loginiai elementai turi po 8 įėjimus (t.y. tiek kiek yra IR elementų išėjimų), kurie sujungti su matricos IR išėjimais. Visi 8 ARBA įėjimai pavaizduoti kaip vienas laidas su 8 programuojamais taškais.

Išėjimo grandinė – tai 4 XOR elementai, kurie, priklausomai nuo programavimo, į išėjimus perduoda tiesioginius arba invertuotus matricos ARBA signalus. Į vieną kiekvienos XOR įėjimą paduoti signalai iš matricos ARBA, į kitus XOR įėjimus po programavimo prijungiamas arba loginis 0 - tuomet išėjimo signalas neinvertuojamas, arba 1 – signalas invertuojamas.

2.5 pav. pavaizduotą PLM galime apibūdinti taip: tai trijų įėjimų PLM, kuri gali atlikti 8 logines sandaugas ir turi 4 išėjimus.

2.2 Programuojamos loginės matricos tipinės funkcinės schemas analizė, veikimo principo, pakopų ir elementų paskirties paaiškinimas

Šioje dalyje apžvelgsime šiuolaikinių programuojamų loginių įtaisų sandarą ir veikimo principą.

Programuojamos loginės matricos sandaros ir principinės schemas analizė, veikimo principo ir pakopų paaiškinimas bus atliekamas kartu dėl didelio integracijos lygio šiame įtaise.

Programuojamos loginės matricos MAX3000 gaminamos įvairiuose korpusuose. Išvadų skaičius priklauso nuo PLM tipo ir gali būti nuo 44 iki 208, bet bendra funkcinė schema ir darbo principas yra toks pats.

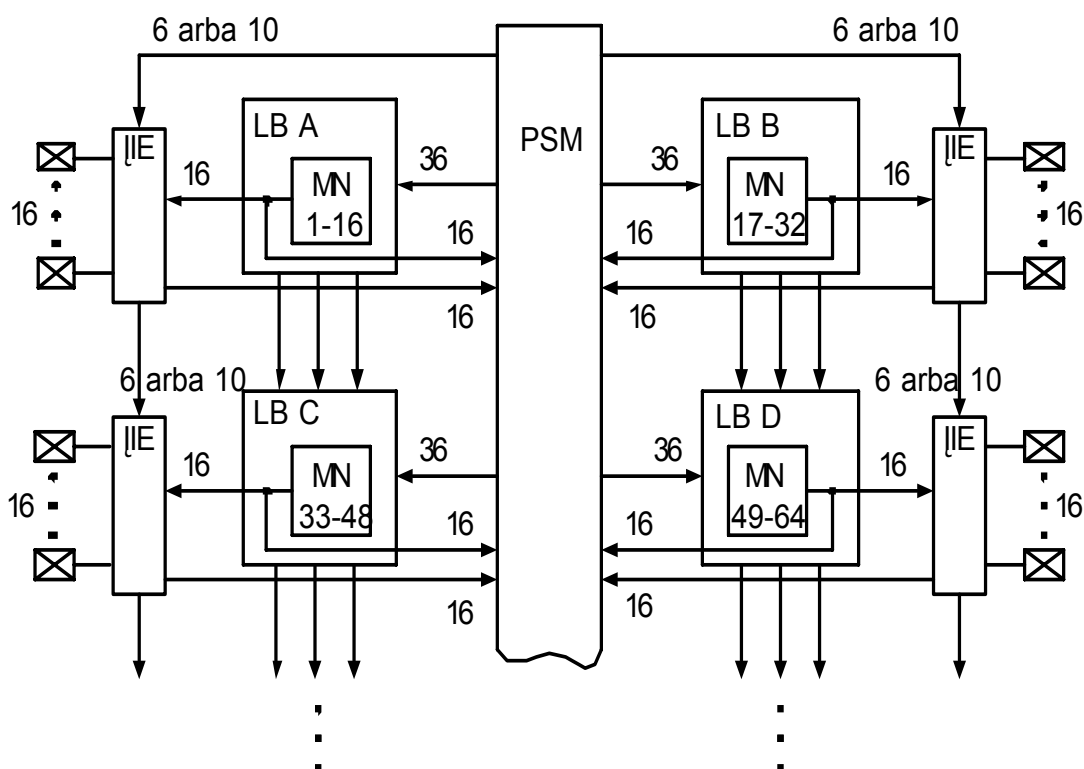
2.6 pav. Parodyta PLM MAX3000 serijos pagrindinių elementų išdėstymas.

Pagrindiniai programuojamos loginės matricos MAX3000 serijos elementai tai:

- loginiai blokai (LB), (LAB, Logic array blocks)
- makronarveliai MN (macrocells)
- loginiai plėstuvai (expanders) (lygiagretus (parallel) ir dalijantis (shareble))
- programuojama sujungimo matrica (PSM), (Programmable interconnect array, PIA)
- įėjimo – išėjimo elementai (IIE),(I/O control block)

PLM MAX3000 serijos turi keturis išvadás, kurie pritvirtinti prie bendrų grandinių (dedicated inputs). Šios bendros grandinės nustato sinchronizacijos numetimą ir į trečią būseną visus

makronarvelius. Taip pat šiuos išvadus galima naudoti kaip įėjimus ir išėjimus, kurie skirti „spartiems“ signalams, kurie apdorojami PLM.



2.6 pav. PLM MAX3000 serijos pagrindinių elementų išdėstymas.

Kaip matome 2.6 pav. PLM MAX3000 serijos architektūros pagrindas yra loginiai blokai. Loginis blokas sudarytas iš 16 makronarvelių sujungtų tarp savęs ir sujungtų su programuojama sujungimo matrica. Kiekvienas loginis blokas turi 36 išvadus nuo PSM.

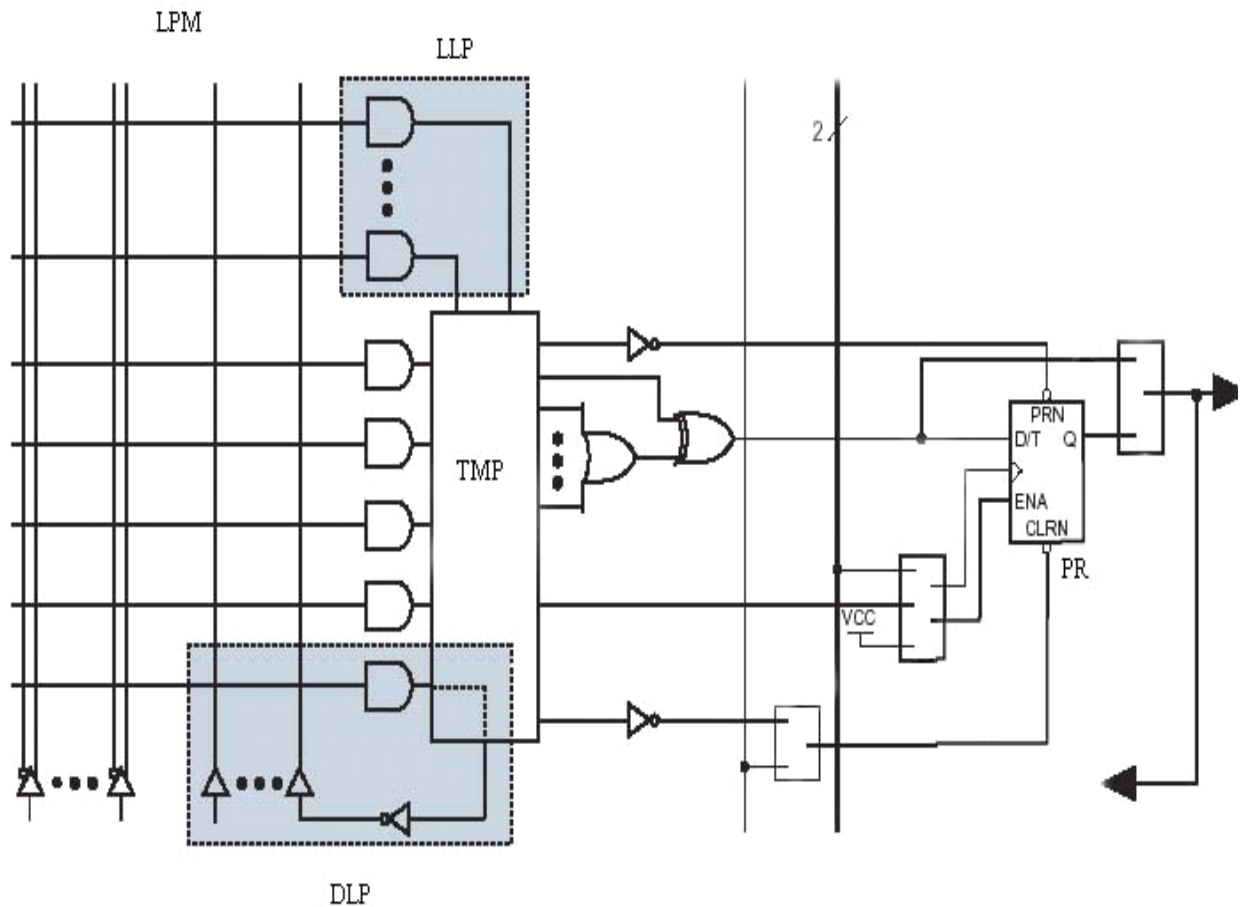
PLM MAX3000 serijos makronarveliai (5.7 pav.) sudaryti iš trijų pagrindinių mazgų:

- lokali programuojama matrica (LPM), (LAB local array)
- termų paskirstymo matrica (TPM), (product-term select matrix)
- programuojamo registro (PR), (Programmable register)

Kombinacinės funkcijos realizuojamos lokaliuose programuojamoje matricoje ir termų paskirstymų matricoje. Jos duoda galimybę sujungti logines sandaugos su ARBA (OR) arba pagal atvirkštinę ARBA (XOR) funkcijas. Taip pat termų paskirstymo matrica duoda galimybę komutuoti makronarvelius, valdymo grandines.

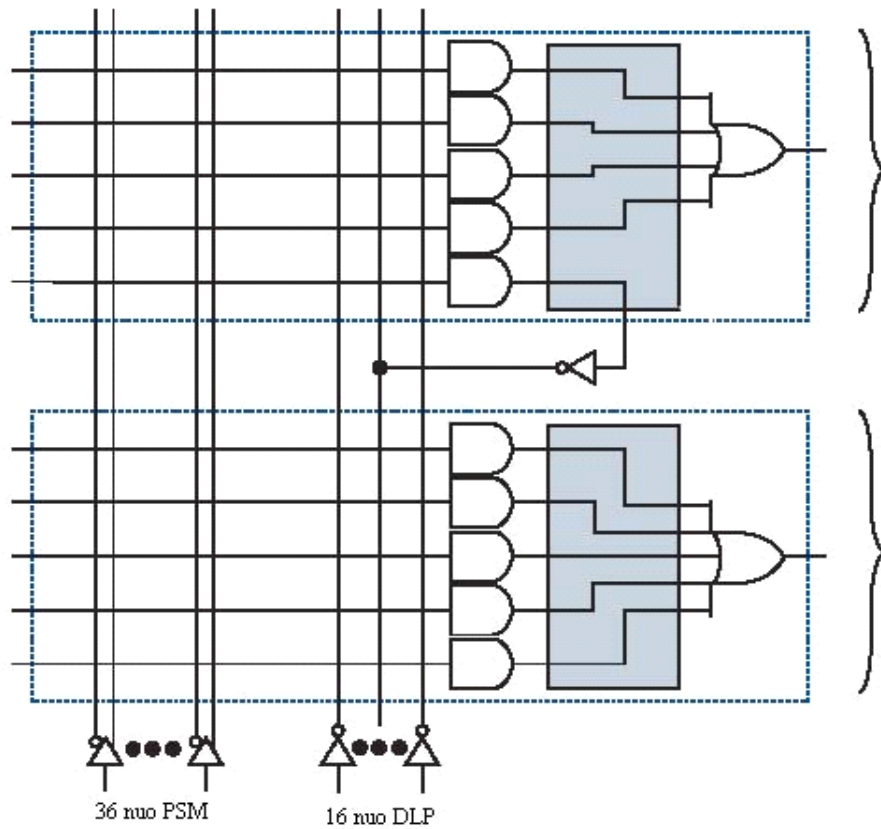
Sinchronizacijos režimas ir trigerio konfigūracija parenkami automatiškai projekto sintezės metu, priklausomai nuo parinkto trigerio tipo aprašant projektą.

PLM MAX3000 serijos yra galimybė parinkti du nepriklausančius vienas nuo kito taktinius signalus. Tai leidžia projektuoti schemas su dviejų fazių sinchronizacija.

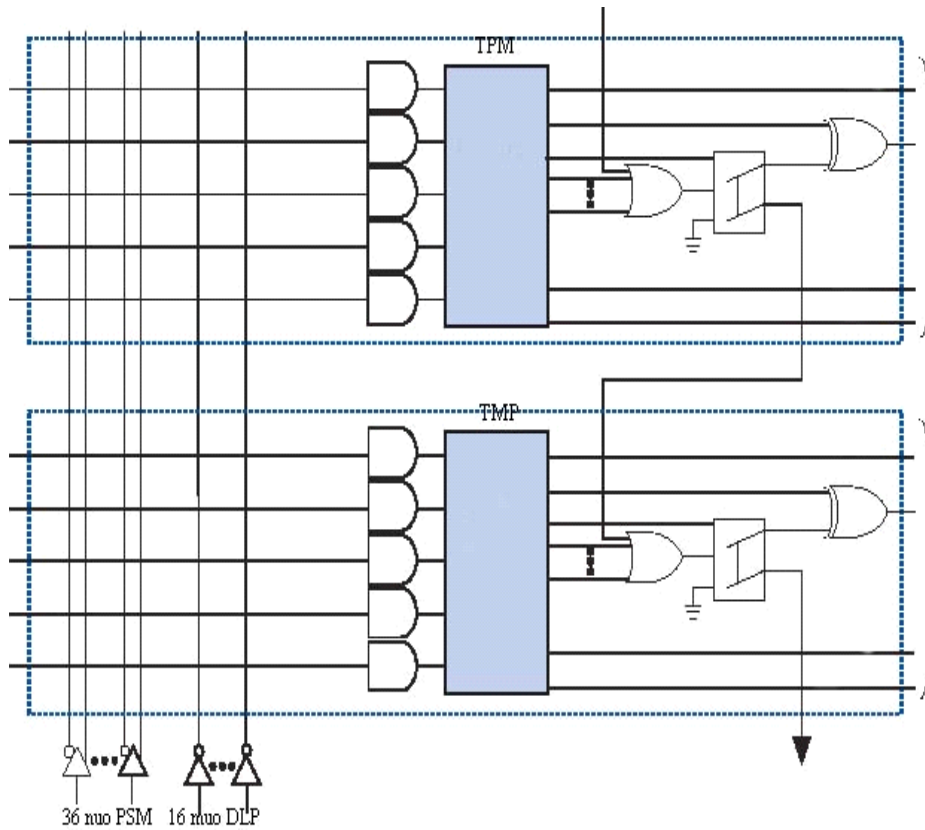


2.7 pav. PLM MAX3000 makronarvelio struktūrinė schema

Loginiai plėstuvai, kai yra daug kintamųjų, naudojami realizuoti logines funkcijas. Dalinantis loginis plėstuvai (2.8 pav.) duoda galimybę sujungti makronarvelius, kurie yra vieno loginio bloko struktūroje. Tokiu atveju dalinantis plėstuvai formuoja termą, kurio inversinė būseną perduodama per termų paskirstymo matricą į lokalią matricą. Perduodamas lygis gali būti panaudojamas su bet koku šio loginio bloko makronarveliu. Kaip matome 2.8 pav. yra 36 signalai nuo programuojamos sujungimo matricos ir 16 inversinių signalų nuo dalijančių loginių plėstuvų, kas duoda galimybę viename loginiame bloke realizuoti funkciją, kuri turi iki 52 termų.

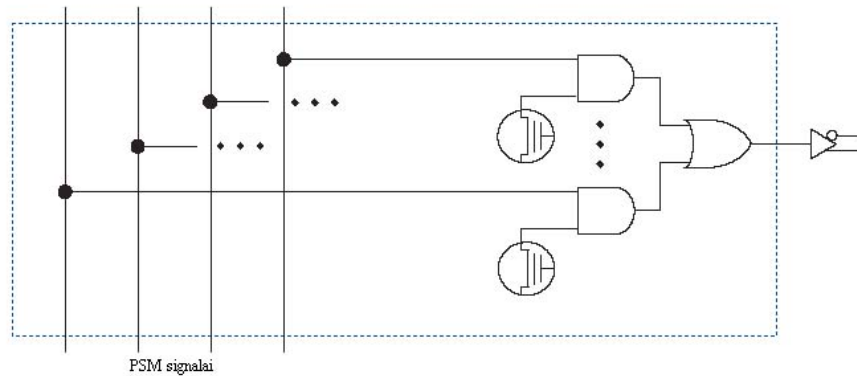


2.8 pav. PLM MAX3000 serijos dalijantis loginis plėstuvas



2.9 pav. PLM MAX3000 serijos lygiagretus loginis plėstuvas

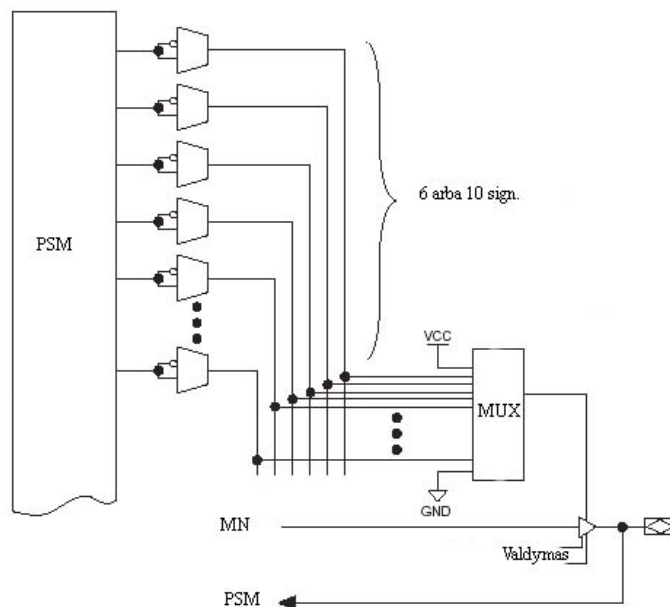
Lygiagretus loginis plėstuvai, pavaizduotas 2.9 pav., duoda galimybę naudoti lokalias gretimas matricas, realizuojant funkcijas, kuriose yra daugiau negu 5 termos. Viena lygiagrečių loginių plėstuvų grandinė gali būti sudaryta iki 4 makronarvelių, realizuojant 20 termų funkciją. Programuojamo paketo MAX+plus II kompiliatorius paliko iki trijų blokų, kuriuose yra ne daugiau negu 5 lygiagretūs loginiai plėstuvai.



2.10 pav. PLM MAX3000 serijos programuojamos sujungimo matricos struktūrinė schema

2.10 pav. parodo PLM MAX3000 serijos programuojamų sujungimo matricų vidinę struktūrinę schemą ir jų išdėstymo tvarką.

Į programuojamą matricą paduodami signalai iš įėjimo – išėjimo elementų, iš loginių blokų atbulinio ryšio. Programavimo metu tik reikalingi signalai paduodami arba grįžta į kiekvieną loginį bloką.



2.11 pav. PLM MAX3000 serijos įėjimo – išėjimo elementų schema

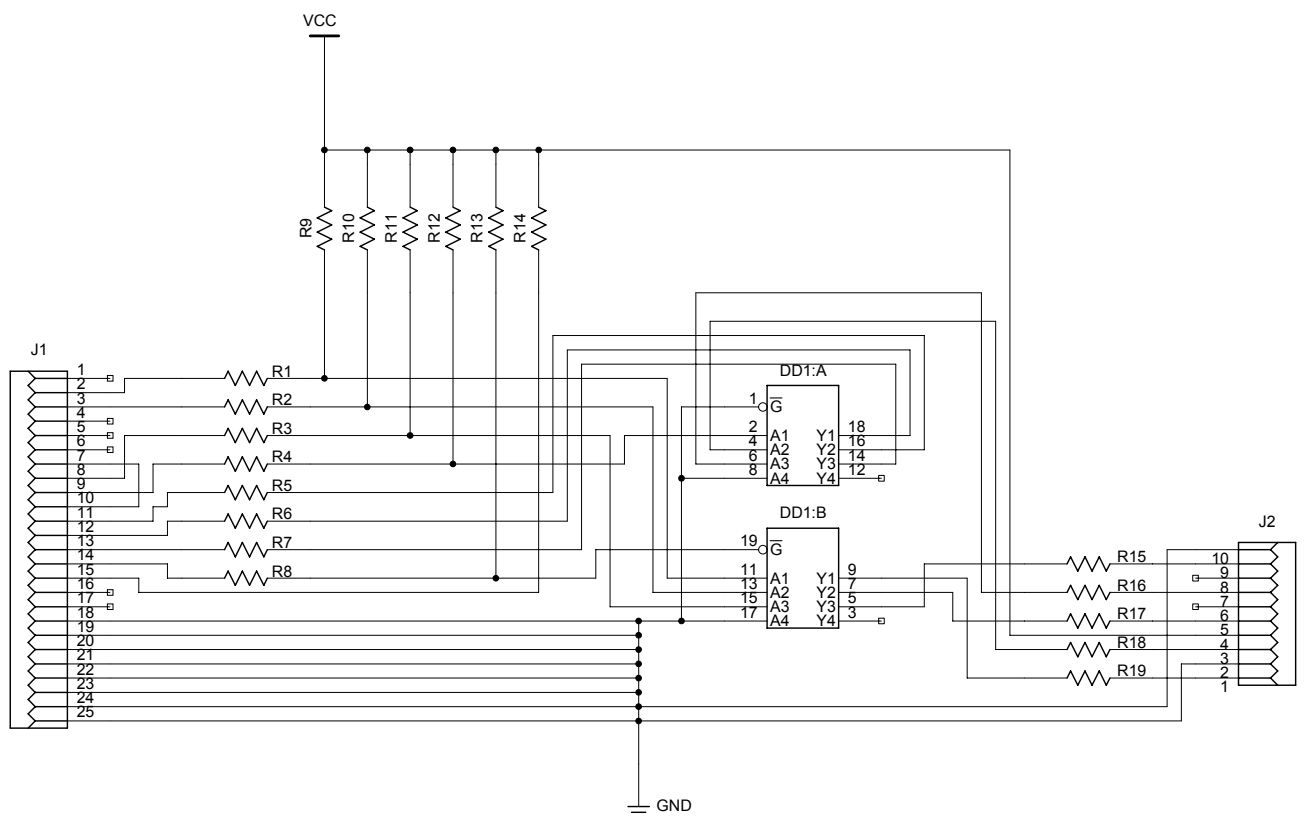
2.11 pav. pavaizduota PLM MAX3000 serijos įėjimo – išėjimo elementų schema, kuri duoda galimybę organizuoti darbo režimą su atviru kolektoriumi ir trečioje būsenoje.

2.3 Programavimo proceso analizė ir principo paaiškinimas

Dažniausiai Alteros bendrovės mikroschemos (MAX7000S, A, B, E, MAX3000, MAX9000) programuojamos su standartinė keturių kontaktų JTAG sąsaja. Programinę įrangą sukūrė konfigūracinis nuoseklumas, pagal kurį programuojama PLM. Visa informacija perduodama per specializuotą laidą (ByteBlaster, BitBlaster arba MasterBlaster). Taip pat PLM galima programuoti naudojant standartinį JTAG testerį arba paprastą sąsają, kuri emuliuoja JTAG komandų nuoseklumą.

ALTERA bendrovė siūlo programavimo metu naudoti konfigūracinį laidą 2.12 pav.) ByteBlaster MV, kuris skirtas programuoti MAX7000S, A, B, E, MAX3000, MAX9000 serijos programuojamas loginės matricas.

Rezistoriai R1÷R8 ir R15÷R19 yra 100Ω (realiai gali būti nuo 50Ω iki 150Ω). Šie rezistoriai nėra privalomi, bet jų panaudojimas apsaugo programuojamą loginę matricą programavimo metu. Rezistoriai R9÷R14 yra 2.2kΩ, bet praktiškai jų varža gali būti nuo 1kΩ iki 3.3kΩ.



2.12 pav. ByteBlaster MV įtaiso principinė elektros schema

Magistralės formavimo mikroschemą galima pritaikyti 74HC244 (galimi analogai: 156AII5 arba 74AC). Taip pat jei programavimas bus atliekamas tik su 5V PLM, galima panaudoti 74ALS ir 74LS mikroschemų serijas (analogai 1533 ir 555 serijų mikroschemos).

3. PLM KONSTRUKCIJA

3.1 ALTERA MAX3000 serijos PLM konstrukcijų apžvalga

Dabar gaminami skaitmeninės technikos įtaisai labai skiriasi nuo tų įtaisų, kurie buvo gaminami 1980-1990 metais. Pirmiausiai tai priklauso nuo to, kad naujos montažo technologijos leido daug kartų sumažinti matmenis. Taip pat atsirado naujo tipo integralinių mikroschemų korpusai, kurie turi mažus atstumus tarp išvadų (0.5 -065 mm). Atsirado naujo tipo mažų gabaritų diskretiniai komponentai ir jungikliai. Padidėjo tikslumas gaminant spausdintąsias plokštes. Visi šitie pakeitimai ir patobulinimai duoda galimybę kurti aukšto lygio sistemas vienoje plokštėje arba viename kristale (system –on –chip). Programuojama loginė matrica taip pat įtaisas su dideliu integracijos lygiu.

MAX3000 serijos PLM pagaminti su 0.30µm technologiniu procesu. Jų darbo temperatūrinis diapazonas yra nuo 0⁰C iki 85⁰C (norint padidinti temperatūrinio diapazono spektrą nuo -40⁰ iki 100⁰ siūloma kreiptis į gamintoją). PLM MA3000 serijos EPM3032A tipų yra 44 išvadų ir jie siūlomi ir su plastikiniu korpusu (Plastic J-Lead Chip), ir su specialiu plonu korpusu (Thin - Quad Flat Pack). Nuo korpuso tipo priklauso tam tikros PLM konstrukcinės savybės (6.1 lentelėje išvardinti atsirandantys skirtumai).

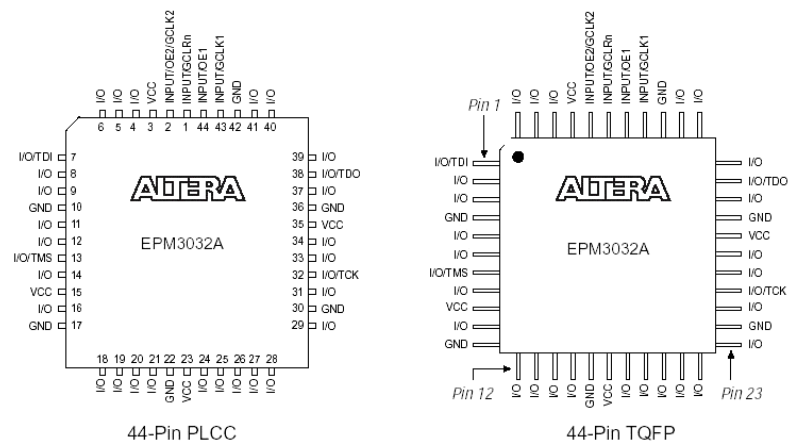
3.1 lentelė. PLM MAX3000 serijos EPM3032A tipo skirtumai (dėl korpuso tipo).

Korpuso tipas	Plastikinis korpusas (Plastic J-Lead Chip)	Specialus plonas korpusas (Thin - Quad Flat Pack)
Išvadų skaičius	44	44
Nominalus ilgis ir plotis (mm)	18x18	12x12
Maksimalus. plotas (mm ²)	312	149
Maksimalus aukštis (mm)	4.57	1.6
Nominalus sujungimų ilgis (mm)	1.27	0.8
Maksimalus sujungimų plotis (mm)	0.53	0.45

Nuo korpuso tipo taip pat priklauso ir išvadų funkcijos (pasikeitė numeracija). 6.2 lentelėje išvardintos visų išvadų paskirtys kiekviename korpuso tipe.

3.2 lentelė. PLM MAX3000 serijos EPM3032A tipo išvadų paskirtis

Paskirtas išvadas	Plastikinis korpusas (Plastic J-Lead Chip)	Specialus plonas korpusas (Thin - Quad Flat Pack)
INPUT/GCLK1	43	37
INPUT/GCLRn	1	39
INPUT/OE1	44	38
INPUT/OE2/GCLK2	2	40
TDI (1)	7	1
TMS (1)	13	7
TCK (1)	32	26
TDO (1)	38	32
GNDINT	22, 42	16, 36
GNDIO	10, 17, 30, 36	4, 11, 24, 30
VCINT (3.3V)	3, 23	17, 41
VCCIO (2.5V arba 3.3V)	15, 35	9, 29
Nėra sujungimo (N. C.)	-	-
Bendras naudojamų įėjimų- išėjimų skaičius (I/O pins)	34	34



3.1 pav. PLM MAX3000 tipo EMP3032A išvadų paskirtis ir numeracija

Šie išvadai gali funkcionuoti ir kaip JTAG sąsaja, ir kaip įėjimo-išėjimo išvadai. Jei išvadai formuojami naudoti JTAG sąsaja, vartoti įėjimo-išėjimo išvadų neįmanoma.

Taip pat būtų gerai įvertinti patarimus konstruojant plokštę su PLM:

- Didelį dėmesį reikia skirti programuojamų loginių matricų įžeminimui. Kaip žinoma, įžeminimas nustatomas kaip ekvipotencialus lygis ir skirtas nustatyti įtampos atskaitos tašką. Projektuojant įžeminimo grandines, reikia minimizuoti įtampos triukšmus, kurie atsiranda pratekant srovei iš atskirų schemų į bendrą įžeminimą. Todėl geriausiai reikia išvengti nuoseklaus įžeminimo (kai skirtingi įžeminimai susijungia viename taške prie įžeminimo).
- Perduodant spartų signalą dideliais atstumais geriausiai panaudoti mažos varžos laidus.
- Projektuojant schemas su PLM geriausiai panaudoti gerą buferinį elementą. Tai gali būti 74HC244, 74HC245 arba panašus elementas.
- PLM naudojimas kuriant impulsines schemas (pvz. generatorius), kurios dirba su išorine talpa ir varža yra nelabai priimtinas. Aišku, tokios schemas gali dirbti, bet PLM nelabai stabiliai dirba su apkrova.
- Labai dažnai schemų kūrėjai parenka filtravimo elementus tik pagal normuotus papildomų mikroschemų lygius, bet reikia įvertinti ir PLM reikalavimus.
- Norint, kad filtrų elementai efektyviai dirbtų, laido ilgis nuo mikroschemos iki talpos elemento turi būti kuo mažesnis.
- Projektuojant plokštes su PLM reikia atkreipti dėmesį į išskiriamą šilumą. Geriausiai panaudoti papildomus metalizacijos sluoksnius.

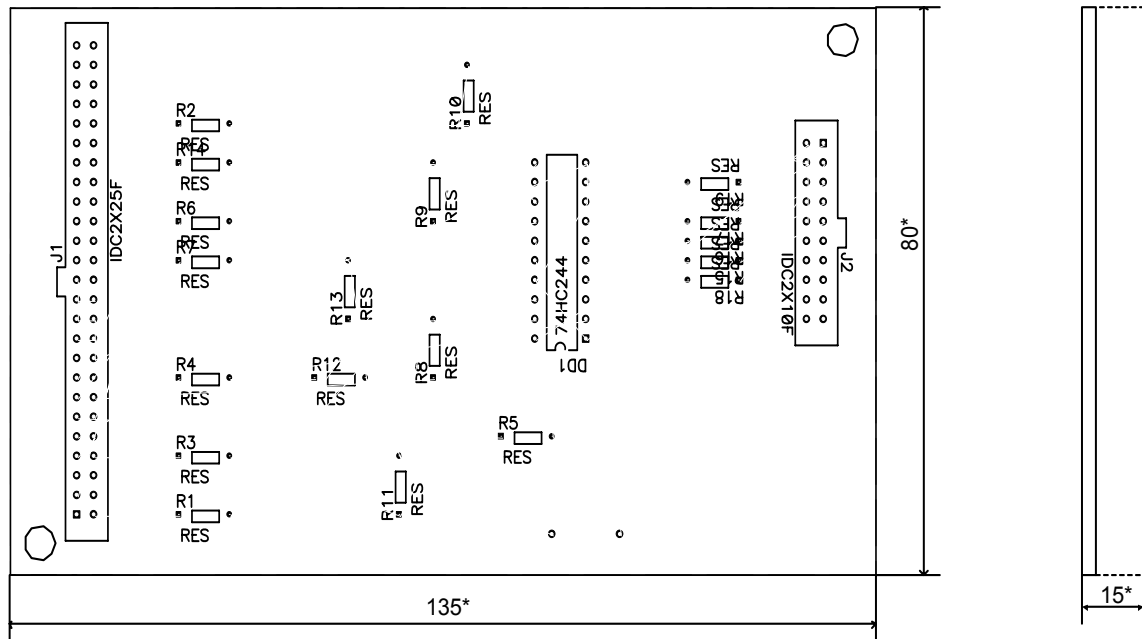
3.2 ByteBlaster MV konstrukcija

Programavimo sistemoje (In-system programmability, ISP) sąvoka yra tuose PLM, kuriuos galima programuoti tiesiogiai plokštėje nenaudojant specialaus programatoriaus. IN-circuit reconfigurability (ISR) procesas duoda galimybę atlikti duomenų siuntimą į PLM schemą, kurios sukurtos pagal SRAM technologiją tiesiogiai, tai yra neišjungiant sistemos maitinimo. ISP ir ISR savybės turi beveik visi PLM, kurie dabar gaminami.

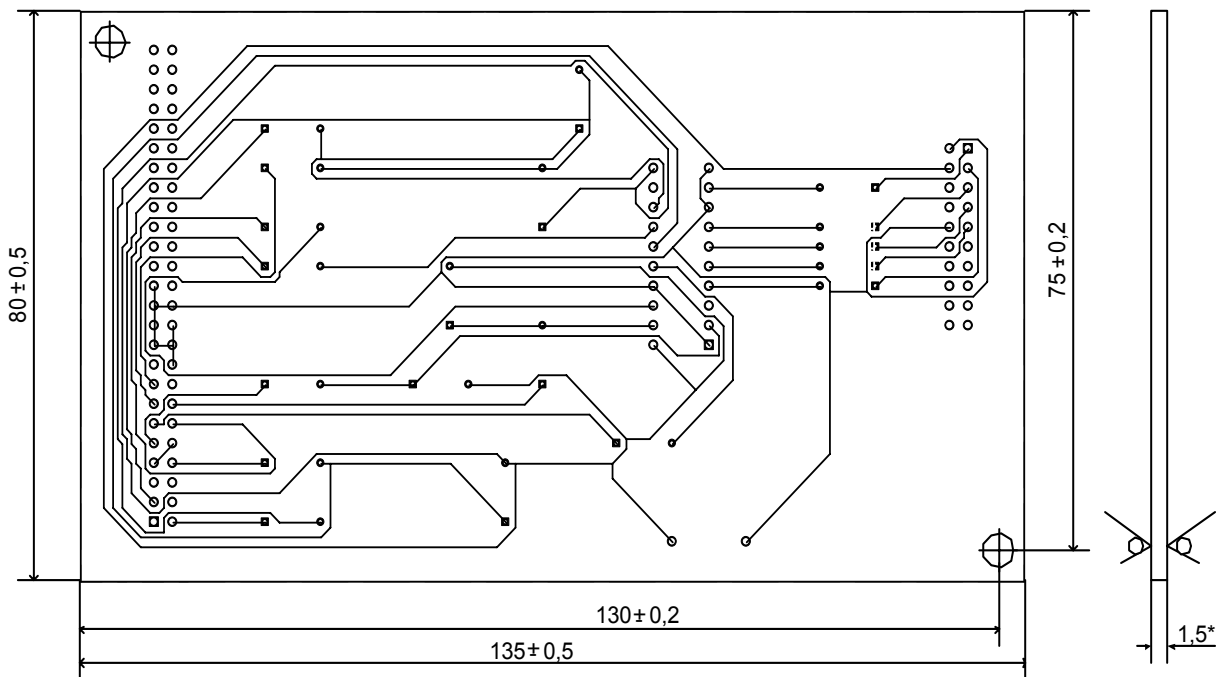
Laido ilgis nuo LTP sąsajos iki ByteBlaster MV laido neturi viršyti $1 \div 1.2$ metrų. Šio laido ALTERA bendrovės siūlomame standartiniame įtaise nėra, bet be jo programavimo procesas yra labai nepatogus. ByteBlaster MV laido ilgis neturi viršyti 0.25 metrų. Dažniausiai naudojamas standartinis 10 gyslų laidas.

ByteBlaster MV plokštės siūloma konstrukcija pavaizduota 3.2 pav. Ši plokštė projektuota naudojant PiCAD programą

3.2 pav. Pavaizduotas suprojektuotos ByteBlaster MV spausdintosios plokštės brėžinys



3.2 pav. ByteBlaster MV spausdintosios plokštės surinkimo brėžinys



6.3 pav. ByteBlaster MV spausdintosios plokštės brėžinys

6.3 lentelėje matome ByteBlaster MV laido kontaktų paskirtį skirtinguose darbo režimuose:

1. pasyvus nuoseklus konfigūracijos režimas (PS mode)
2. programavimo režimas su JTAG sąsaja (JTAG mode).

3.3 lentelė. ByteBlaster MV laido kontaktų paskirtis

Sąsajos ByteBlaster kontaktas	PS Mode		JTAG mode	
	Signalas	Paskirtis	Signalas	Paskirtis
1	DCLK	Taktinis signalas	TCK	Taktinis signalas
2	GND	Ižemintas	GND	Ižemintas
3	CONF_DONE	Konfigūracijos baigimo valdymas	TDO	Informacija nuo PLM
4	VCC	Maitinimo įtampa	VCC	Maitinimo įtampa
5	nCONFIG	Konfigūracijos kontrolė	TMS	JTAG automato kortelė
6	-	Neprijungtas	-	Neprijungtas
7	nSTATUS	Konfigūracijos būseną	-	Neprijungtas
8	-	Neprijungtas	-	Neprijungtas
9	DATA0	Informacija į PLM	TDI	Informacija į PLM
10	GND	Ižemintas	GND	Ižemintas

4. PROJEKTAVIMO SISTEMA MAX+plus II

4.1 Pagrindiniai projekto sukūrimo principai

Projektavimo sistemos MAX+plus II pavadinimas atsirado nuo abreviatūros – **Multiple Array MatriX Programmable Logic User System**. Sistema MAX+plus II, turi galimybes įvesti projektą, sukompiliuoti, paderinti ir tiesiogiai programuoti įtaisus.

Naujo projekto sukūrimo principus nuo koncepcijos iki pabaigos galima įsivaizduoti taip:

1. Naujo projekto failo arba kelių failų sukūrimas, kurių struktūrinis tipas yra hierarchinis. Tam tikslui galima panauduoti grafinį, signalinį arba tekstinį redaktorius.
2. Priskirti failui projekto vardą.
3. Priskirti tam tikrą PLM projektui.
4. Atidaryti kompiliatoriaus langą Compiler ir pasirinkus mygtuką Start pradėti kompiliacijos procesą. Jei vartotojas norės, galima prijungti modulį, kuris išrinkinės projekto Timing SHF Extractor laiko parametrus ir sukurs failą, kurį bus galima panaudoti atliekant laiko modeliavimą.
5. Jei kompiliacijos procesas buvo sėkmingas, galima atlikti laiko analizę, tam tikslui reikia atlikti šiuos veiksmus:
 - Norint atlikti užlaikymų laikinę analizę, reikia atidaryti Timing Analyzer langą, išrinkti analizės režimą ir paspausti mygtuką Start;
 - Norint atlikti simuliaciją, pradžioje reikia, panaudojant signalinį redaktorių, sukurti testavimo vektorius testavimo faile (.scf), arba panaudojant tekstinį redaktorių užduoti testavimo vektorius vektoriniame faile (.vec). Po to atidaryti Simulator langą ir paspausti mygtuką Start.
6. Atidaryti programatoriaus langą Programmer ir išsirinkti vieną iš dviejų programavimo būdų: panaudojant programatorių MPU (Master Programming Unit) arba prijungti pakrovimo įtaisus BitBlaster, ByteBlaster arba FLEX Download Cable prie įtaiso, kuris bus programuojamas.
7. Parinkti mygtuką Program programuoti įtaisus su EPROM arba EEPROM atminties tipu. Jei įtaisas turi SRAM tipo atmintį, reikia atlikti įtaiso derinimą, paspaudus mygtuką Configure.

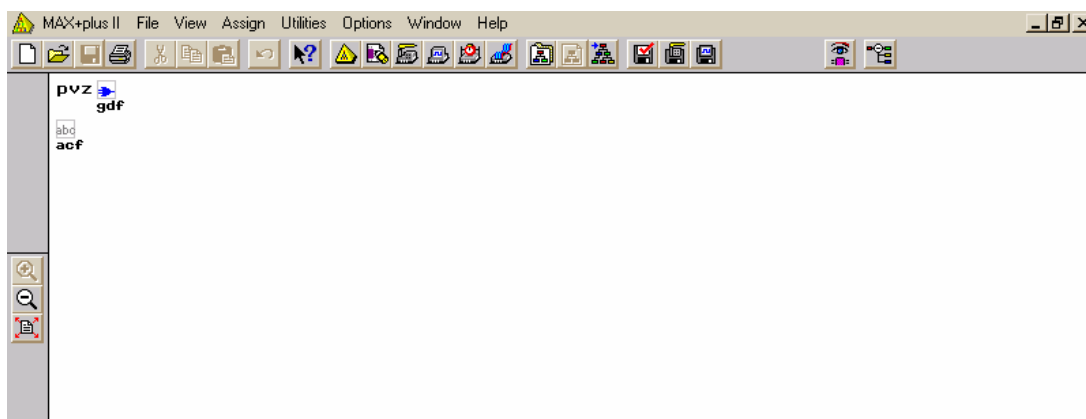
4.2 Redaktorių paaiškinimas

Programinis paketas MAX-plus II turi 11 priedų ir pagrindinę valdymo programą. Įvairūs priedai, kurie duoda galimybę sukurti projektą, gali būti aktyvuoti bet kuriuo momentu panaudojant pelę arba komandas iš meniu. Tuo pačiu momentu gali dirbti ir kiti programos priedai, pavyzdžiui

kompiliatorius, simulatorius, sinchronizacijos analizatorius ir programatorius. Tos pačios įvairių priedų komandos dirba vienodai, kas lengvina loginio dizaino sukūrimą.

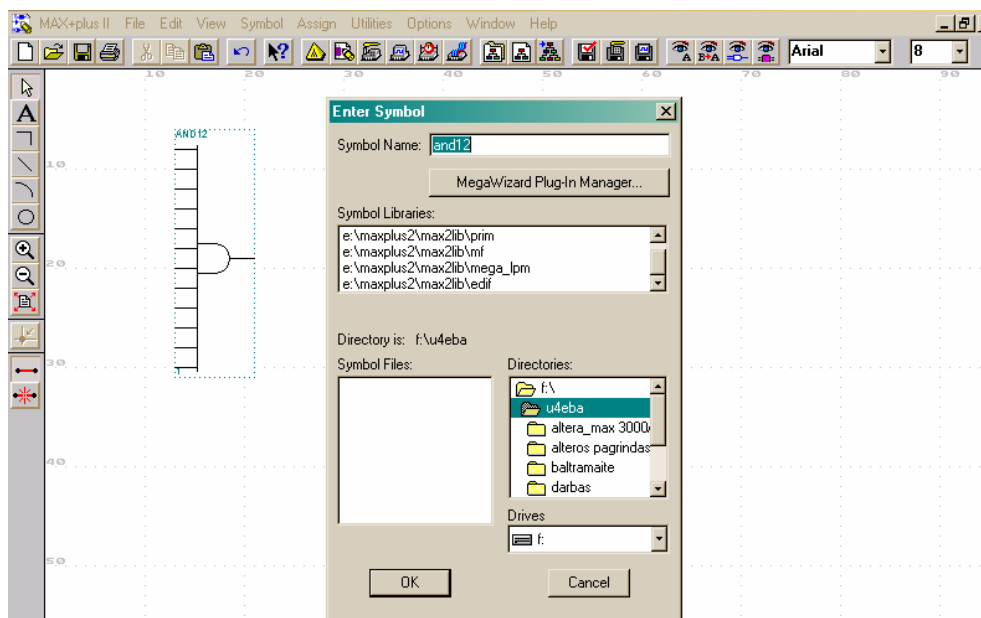
Kiekvienas priedas programiniame pakete MAX+plus II turi savo funkcijas ir privalumus:

Hierarchy Display (4.1 pav.) – tai hierarchijos apžvalgos priedas, kuris atvaizduoja esančią hierarchijos failų struktūrą medžio su šaknimis pavidalu. Medžio šaknys – tai prijungti projektai prie pagrindinio projekto.



4.1 pav. Hierarchy Display priedo langas

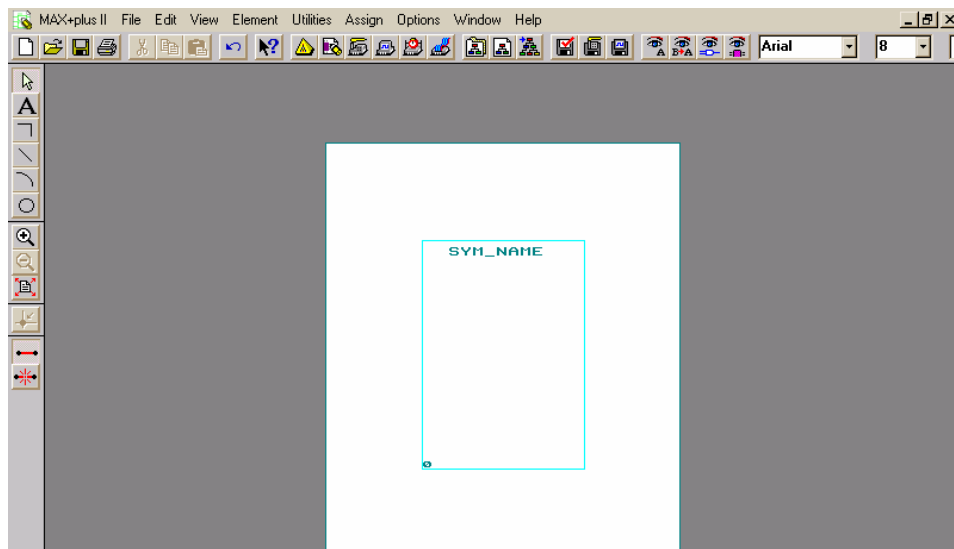
Graphic Editor (4.2 pav.) – tai grafinis redaktorius, kuris leidžia kurti scheminį loginį projektą, kuris bus realiai atvaizduojamas ekrane WYSIWYG. Projekto grafiniai failai (.gdf) arba scheminiai OrCAD failai (.sch) sukurti šiame grafiniame redaktoriuje gali turėti bet kokia paprastų simbolių kombinaciją. Simboliai gali būti bet kokių projektų failų.



4.2 Craphic Editor priedo langas

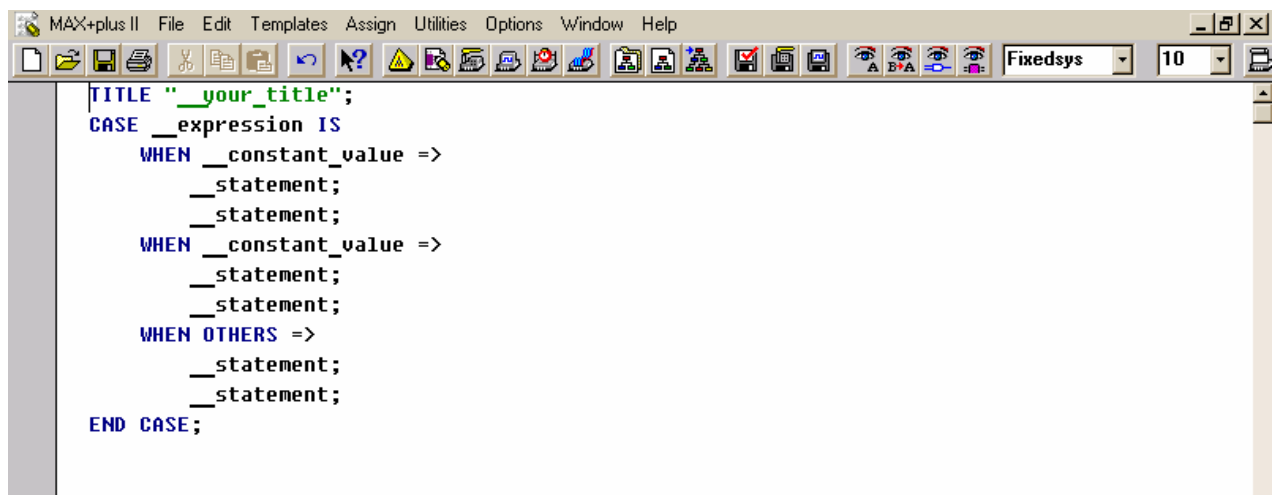
Symbol Editor (4.3 pav.) – tai simbolių redaktorius, kuris leidžia redaguoti esančius simbolius ir kurti naujus. Simbolinis failas turi tokį pat vardą kaip ir projektas. Komanda Creat

Default Symbol kuri yra visose redaktoriuose sukuria simbolių bet kuriam projekto failui. Simbolinis redaktorius gali kurti ir redaguoti kontaktus ir jų vardus, užduoti parametrus.



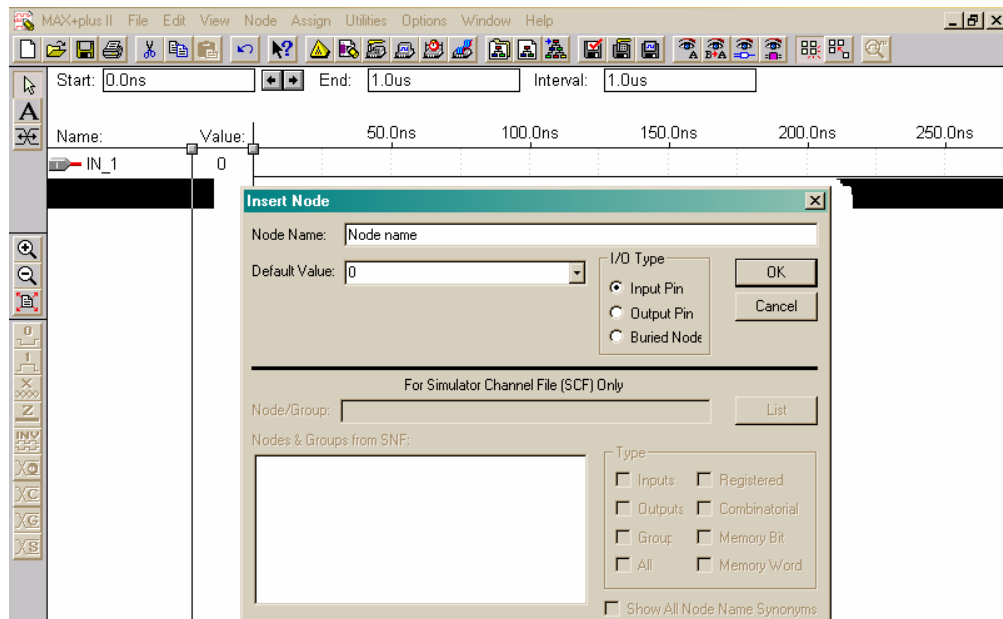
7.3 pav. Symbol Editor priedo langas

Text Editor (7.4 pav.) – tai tekstinis redaktorius, kuris leidžia kurti ir redaguoti tekstinius failus su loginiu dizainu, kurie parašyti AHDL, VHDL ir Verilog HDL kalba. Tekstiniame redaktoriuje taip pat galima dirbti su ASCII formato failu. Galima redaguoti .acf failus ir nustatyti kompiliatoriui ir laiko analizatoriui norimą konfigūraciją.



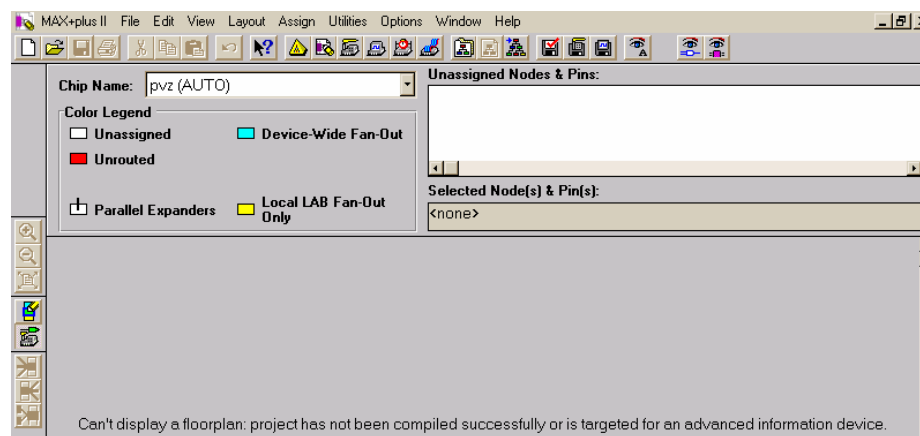
4.4 pav. Text Editor priedo langas

Waveform Editor (4.5 pav.) – tai signalinis redaktorius, kuris atlieka iškart dvi funkcijas: tai dizaino kūrimo redaktorius ir redaktorius, kuriame įvedinėjami testavimo vektoriai norint peržiūrėti testavimo rezultatus. Vartotojas gali kurti projekto signalinius failus .wdf, kuriuose yra laiko diagramos. Šios diagramos aprašo projekto logiką. Failai .scf testuoja projekto darbą.

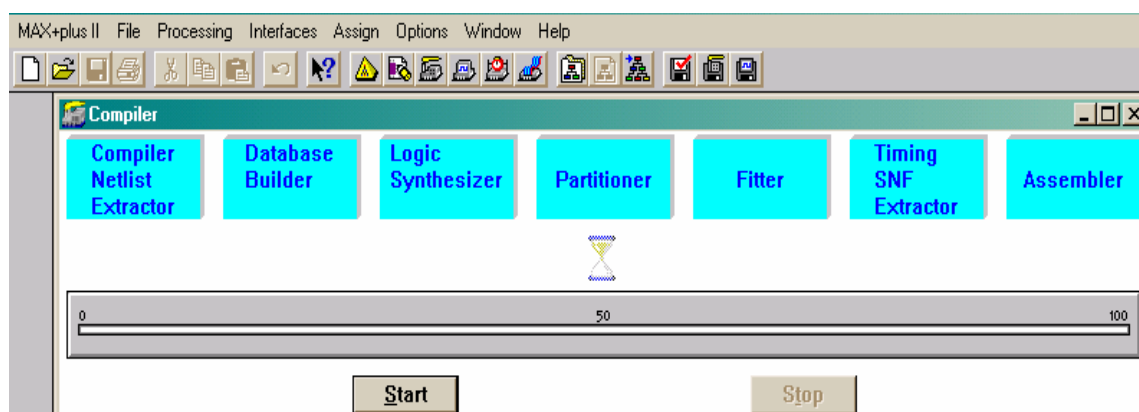


4.5 pav. Waveform Editor priedo langas

Floorplan Editor (4.6 pav.) – demonstracijos redaktorius, kuris leidžia grafiniu būdu sudaryti priskyrimus įtaiso kontaktams ir loginių elementų resursams.



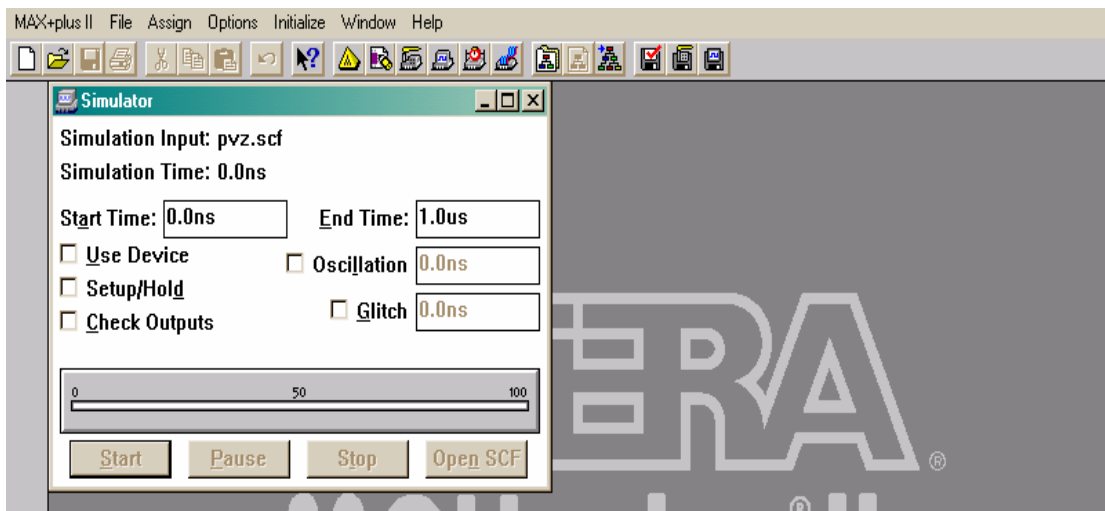
4.6 pav. Floorplan Editor priedo langas



4.7 pav. Compiler priedo langas

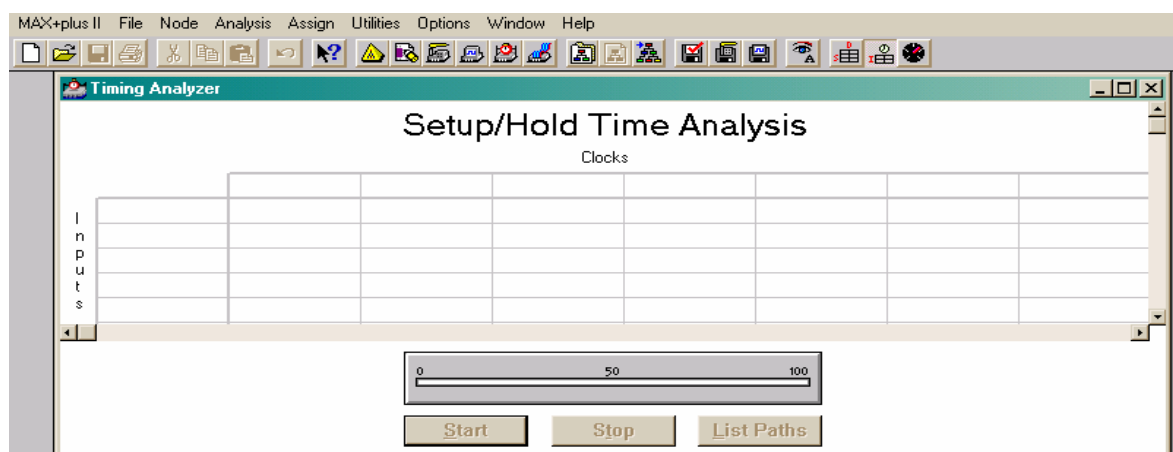
Compiler (4.7 pav.) – tai kompiliatorius, kuris apdoroja loginius projektus. Kompiliavimo proceso metu galima pasirinkti filtrus ir laiko patikrinimo metodiką.

Simulator (4.8 pav.) – simulatorius, kuris leidžia testuoti loginės operacijos ir projektuojamos loginės grandinės vidinę sinchronizaciją.



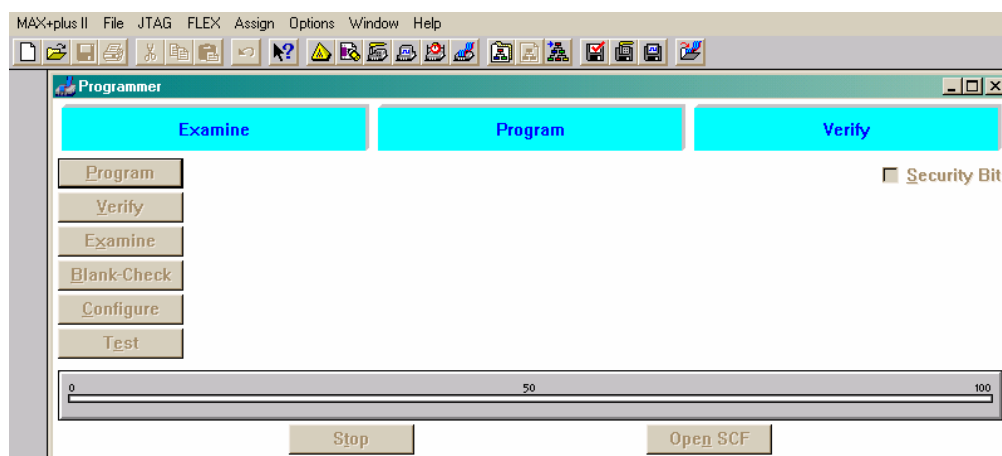
4.8 pav. Simulator priedo langas

Timing Analyzer (4.9 pav.) – laiko analizatorius, kuris analizuoja projektuojamos loginės grandinės darbą po to, kai ji buvo sintezuota ir optimizuota kompiliatoriaus.



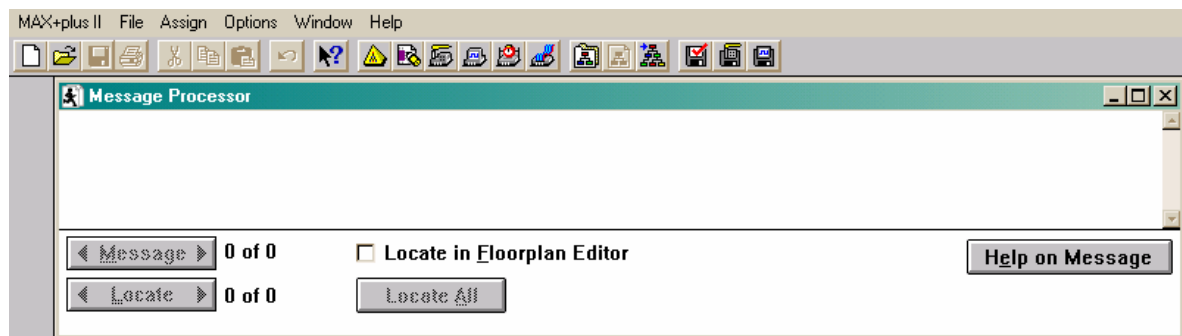
4.9 pav. Timing Analyzer priedo langas

Programmer (4.10 pav.) – programatorius, kuris leidžia programuoti, derinti, atlikinėti testavimą Alteros firmos PLM.



4.10 pav. Programmer priedo langas

Message Procesor (4.11 pav.) – pranešimų generatorius, kuris išveda į ekraną pranešimus apie klaidas ir kitus informacinio tipo pranešimus.



4.11 pav. Message Procesor priedo langas

4.3 Darbas su projektu

Prieš pradėdant dirbti MAX+plus sistema reikia suprasti skirtumą tarp projekto failų ir pagalbinių projektų.

Projekto failas – tai grafinis, tekstinis arba signalinis failas, kuris buvo sukurtas panaudojant tam tikrą projektavimo sistemos MAX+plus II redaktorių. Failas, kuriame užrašyti loginiai dėsniai kompiliuojamas kompiliatoriais. Kompiliatorius gali automatiškai apdoroti įvairius projekto failus: grafinius projekto failus (.gdf); tekstinius projekto failus, kurie buvo parašyti AHDL kalba (.tdf); signalinius projekto failus (.wdf); VHDL kalbos projekto failus (.vhd); Verilog kalbos projekto failus (.v); programinio paketo OrCAD scheminius failus (.sch); EDIF programinio paketo įėjimo failus (.edf); failus, kurių formatas yra Xilinx Netlist paketo (.xnf); Alteros projekto failus (.adf); skaitmeninio automato failus (.smf).

Pagalbiniai failai – tai failai, kurie susiję su MAX+plus II projektu, bet jie nėra projekto hierarchinio medžio dalis. Dauguma tokių failų neturi kūrimo projekto logikos. Kai kurie iš jų kuriami automatiškai kokių nors projektavimo sistemos priedų arba juos kuria pats vartotojas. Tokių failų pavyzdys yra priskyrimo ir derinimo failai (.acf), simboliniai failai (.sym), ataskaitos failai (.rpt) ir testavimo vektorių failai (.vec).

Projektas sudarytas iš visų failų, kurie yra jo hierarchinėje struktūroje. Projekto vardas – tai failo vardas, kuris yra projekto viršutiniame lygyje. Sistema MAX+plus II atlieka kompiliavimą, testavimą, sinchronizacijos analizę ir iškart viso projekto programavimą, nors vartotojas gali tuo pačiu laiku redaguoti failus kitame projekte. Kiekvienam projektui geriausiai kurti atskirą vietą darbui. MAX+plus sistemoje visi projekto kūrimo instrumentai lengvai pasiekiami. Kūrimas pagreitinamas panaudojant standartines funkcijas, modulius (LMP) ir 74 serijų mikroschemų funkcijas.

Hierarchinėje projekto struktūroje bet kuriame lygyje yra galimas mišrus .gdf, .tdf, .vhd, .v, .edf, .scf failų panaudojimas. Bet failai .wdf, .xnf, .adf, .smf turi būti hierarchiniame apatiniame lygyje arba jie turi būti vieninteliai projekto failai.

Visuose MAX+plus II prieduose yra galimybė panaudojant komandas iš Assign lentelės įvedinėti ir redaguoti parametrus, kurie valdo kompiliacijos procesą. Vartotojas gali daryti priskyrimą projektui nepriklausomai nuo to, ar aktyvuotas koks nors projekto failas arba priedo langas. Galimi įvairūs priskyrimo tipai:

Clique assignment užduoda, kokios loginės funkcijos turi būti paliktos tame pačiame loginiame bloke LAB, atminties narvelių bloke EAB, vienoje eilėje arba įtaise.

Chip assignment užduoda kokios loginės funkcijos turi būti realizuotos tame pačiame įtaise, jei projektas bus padalintas į keliolika įtaisų.

Pin assignment priskiria realiam kontaktui įėjimą arba išėjimą vienos loginės funkcijos.

Location assignment priskiria vieną loginę funkciją konkrečiam kristalo narveliui.

Probe assignment priskiria unikalų vardą loginės funkcijos įėjimui arba išėjimui.

Connected pin assignment užduoda dviejų arba daugiau kontaktų išorinį sujungimą schemeje.

Local routing assignment priskiria iššakojimo koeficientą loginio elemento išėjimui, kuris yra tame pačiame bloke LAB. Jei naudojami bendri vietiniai iššakojimai, galimas kito bloko LAB panaudojimas.

Device assignment užduoda programuojamos loginės matricos tipą, kurioje bus realizuojamas projektas.

Logic option assignment valdo loginius sujungimus atskirų loginių funkcijų kompiliacijos metu.

Timing assignment valdo loginių sujungimų ar atskirų loginių funkcijų derinimui, norint gauti laiko parametrus.

Galima suderinti kompiliatorių taip, kad jis naudotų bendrus nustatymus visiems įtaisams. Norint rezervuoti papildomas galimybes, galima užduoti, kiek procentų (kontaktai/loginiai elementai) turi būti neišnaudota kompiliacijos metu.

Panaudojant Global Project Parameters komanda galima įvesti vardus ir globalius parametrus, kurie bus naudojami kompiliatorių parametrų visoms funkcijoms projekte, kurios turi užduotus parametrus.

Global Project Timing Requirements duoda galimybę įvesti projekto globalius sinchronizacijos parametrus, nustatyti bendrus laiko užlaikymo parametrus.

Global Project Logic Synthesis nustato globalius parametrus kompiliatoriui projekto loginiame sinteze.

Visi penki MAX+plus II redaktoriai ir trys kūrimo redaktoriai turi bendras funkcijas, tokias kaip sukūrimas ir failo atidarymas. Taip pat redaktorių priedai gali sudaryti simbolių failus, failus su funkcijomis (include failai), mazgų paiešką, laiko analizę, teksto fragmentų paiešką ir kitas windows tipo paprasčiausias operacijas. Visa tai palengvina projektavimo darbą ir labai teigiamai charakterizuoja projektavimo sistemą MAX+plus II.

5. PRAKTINIAI PLM PROGRAMAVIMO DARBAI

5.1 Darbas su AHDL programine kalba

Programinė kalba AHDL buvo sukurta Alteros bendrovėje ir yra skirta aprašyti kombinacines grandines, loginius įtaisus, grupines operacijas, skaitmeninius automatus (state machine) ir reikšmių lenteles, įvertinant Alteros bendrovės programuojamų loginių matricų architektūrinės savybes. AHDL kalba pilnai integruojama į MAX+plus II projektavimo sistemą. AHDL kalbos aparatūros aprašymo failai yra .TDF formato (Text Design File). Norint sukurti .TDF failą, galima naudoti MAX+plus II tekstinį redaktorių arba kitus programinius paketus. Sukurtas projektas kompiliuojamas, derinamas ir naudojamas formuoti kitą programavimo failą arba programuoja Alteros firmos PLM.

Programinės kalbos AHDL elementai ir operatoriai yra labai universalūs ir patogūs būdas aprašyti skaitmeninių įtaisų funkcionavimo algoritmus. Su juo galima kurti hierarchijos tipo projektus, kai visas projektas parašytas su AHDL kalba, taip ir kada naudojamų kitų tipų projekto failai.

MAX+plus II projektavimo sistema duoda galimybę automatiškai sukurti komponento simbolį, kuriuo algoritmas aprašomas su AHDL kalba. Po to šį failą galima integruoti į scheminį aparatūros aprašymą (GDF tipo failas). Projektavimo sistemoje yra tokio tipo paruošti failai. Tai .inc tipo failai kurie naudojami su INCLUDE operatoriumi.

Dalijant resursus, kūrėjas gali naudotis tekstinio redaktorių komandomis arba AHDL kalbos šablonais. Taip pat kūrėjas gali patikrinti sintaksę (check komanda) arba atlikti pilną kompiliacijos procesą. Kompiliavimo metu sintaksė tikrinama automatiškai.

Dirbant su AHDL programine kalba geriausiai vykdyti Alteros bendrovės „Auksinę darbo metodiką“ (Golden Rules). Šių taisyklių naudojimas duoda galimybę efektyviai taikyti AHDL kalbą ir išvengti daugelio galimų klaidų:

- Reikia naudoti standartinius formatus, kurie suderinti su AHDL kalba. Tai leis visada perskaityti projektą ir sumažins klaidų kiekį.
- Nekreipiant dėmesio į tą faktą, kad AHDL kalboje nėra skirtumo tarp didelių ir mažų raidžių, ALTERA bendrovė siūlo naudoti dideles raides pagrindiniuose žodžiuose.
- Nereikia naudoti sąlygų operatorių IF šablonų, jei galima naudoti parinkimo operatorių CASE.
- TDF failo eilutė gali būti ne ilgesnė negu 255 simbolių. Bet geriausiai naudoti tokio ilgio eilutes, kurios telpa į ekraną. Eilutė baigiasi paspaudus Enter mygtuką.

- Naują eilutę galima pradėti bet kurioje laisvoje vietoje. Pagrindinės kalbos konstrukcijos skiriamos su tuščia erdve.
- Pagrindiniai žodžiai, vardai ir skaičiai turi būti skiriami panaudojant specialius simbolius arba operatorius.
- Komentarai turi būti skiriami su procento ženklu (%). Komentaruose gali būt bet koks simbolis, neskaitant % simbolio. MAX+plus sistemos kompiliatorius ignoruoja visus komentarus.
- Sujungiant paprasto tipo modelius, reikia naudoti tik leidžiamo tipo sujungimus, nes ne visi elementai gali būti sujungiami.
- Geriausiai naudoti tik EXPDFF, EXPLATCH, NANDLTCH ir NORLTCH tipo makrofunkcijas, kurie yra MAX+plus II sistemoje. Pačiam geriausiai nekurti susijungimo struktūros. Taip pat geriausiai nenaudoti šių funkcijų kartu. Šios funkcijos visada turi būti skiriamos su LCELL funkciniu šablonu.

Programuojant LPM su AHDL kalba taip pat geriausiai naudoti bendrą „Auksinę metodiką“:

- Jeigu daug dviejų krypčių arba išėjimų išvadų surišti tarp savęs, kūrėjas negali naudoti PIN CONNECTION funkcijos, norėdamas sujungti išvados modeliujant funkcionaliai su aparatinium palaikymu arba funkcionaliai testuojant.
- Nėra prasmės kurti funkcijų prototipų paprastoms funkcijoms. Bet kūrėjas gali iš anksto numatyti paprastas funkcijas, norint keisti įėjimų aktyvacijos eilę TDF faile.
- Nereikia redaguoti .FIT formato failo. Jei kūrėjas nori redaguoti projekto paskirtį, reikia pradžioje nukopijuoti failą TDF formate ir tik po to padaryti atvirkštinį priskyrimą su Project Back – Annotate komanda. Po to geriausiai atlikti redagavimą su Chip to Device, Pin/LC/Chip ir Enter Assignments komandų pagalba.
- Jeigu kūrėjas nori pakrauti registrą, panaudojant bendrą taktinį signalą Clock, ALTERA bendrovė rekomenduoja (kai registras pakrautas) naudoti vieną iš Enable tipo trigerių (DFFE, TFFE, JKFFE arba SRFFE) valdyti Clock Enable įėjimą.
- Pradedant dirbti su projekto naujo failo, reikia iškart nurodyti PLM seriją, kuriems skirtas projektas, panaudojant Family konstrukciją. Tai daroma norint turėti galimybę pasinaudoti makrofunkcijomis, kurios yra specifinės kuriai nors serijai. Jei kūrėjas nenurodys PLM serijos tipo, programinis paketas dirbs su tomis makrofunkcijomis, kurios skirtos praeito projekto PLM.
- Norint patikrinti projekto logikos patikimumą, reikia naudoti Design Doctor opciją kompiliacijos metu.

- Alteros bendrovės siūlomos loginio sintezės stiliai yra įvairūs ir priklauso nuo įtaisų serijos. Tai duoda galimybę efektyviai naudoti kiekvienos serijos architektūrinius bruožus. Pakeičiant programuojamos serijos tipą, reikia būtinai patikrinti naujo stiliaus nustatymus.

Dirbant MAX+plus II sistemoje taip pat yra savos taisyklės:

- Darbo pradžioje su nauju projektu rekomenduojama iškart užduoti jo vardą kaip ir naujo projekto panaudojant Project Name komandą. Šiuo atveju projektas bus lengvai sukompiliuotas. Vėliau projekto kūrėjas visada galės pakeisti projekto vardą.
- Pereinant nuo vieno projekto failo prie kito, geriausiai visada naudoti projektavimo sistemos MAX+plus II hierarchines galimybes. Norint atidaryti žemesnio lygio failą, reikia atidaryti aukštesnio lygio failą ir panaudojant Hierarchy Display langą (arba Hierarchy Down) atidaryti žemesnio lygio failą. Jeigu kūrėjas parenka Open arba Retrieve projektavimo sistemą, laikoma, kad tai aukščiausio lygio failas ir visi resurso paskirstymai, įtaisų paskirstymai bus priskiriami prie šios naujos hierarchijos, o ne prie pagrindinio projekto.
- Jeigu kūrėjas sukūrė pagalbinį failą kokiam nors projektui, šio failo piktograma atsiras Hierarchy Display lange tik tuo atveju, jei ji turi toki pat vardą, kaip ir pats projektas.
- Nerekomenduojama redaguoti projektavimo sistemos MAX+plus II sisteminių failų. Taip pat nereikia redaguoti projekto failų .prb, .hif arba .ini formato.
- Jeigu projekto kūrėjas nori pakeisti projekto vardą arba pagalbinio failo vardą, geriausiai taikyti Save as ... komandą. Nerekomenduojama keisti failo vardą ne iš MAX+plus II programinio paketo.
- Atlikus projektavimą, rekomenduojama atlikti projekto glaudimą panaudojant komandą Project Archive, sukuriant rezervinę kopiją visam projektui. Tuo atveju rezervinė kopija bus apsaugota nuo tolesnių projekto redagavimų.

AHDL programinėje kalboje galima naudoti: skaičius (dvejetainę, aštuntainę, dešimtainę ir šešiolyktainę sistemas), konstantas (priskyrimas vykdomas panaudojant komandą Constant), kombinacinius logikos tipo užrašymus, realizuoti logines lygybes ir dirbti su integruotais operatoriais.

5.2 Darbas su VHDL programine kalba

Programinė kalba VHDL – tai formalus užrašymas, kuris gali būti panaudotas visuose skaitmeninių grandinių projektavimo etapuose. Ši galimybė atsiranda dėl to, kad ši programinė kalba lengvai suprantama kaip ir įtaisui, taip ir žmogui. Programinė kalba gali būti panaudojama visuose projektavimo etapuose, testuojant aparatūrą, perduodant duomenis apie projektą, modifikuojant. Visuose atvejuose programinė kalba VHDL bus universalesnė. Ši programinė kalba leidžia aprašyti kaip ir skaitmeninės grandinės darbą, taip ir struktūros tipą.

Programinė kalba VHDL naudojama daugelyje sistemų, modeliuojant skaitmenines grandines, projektuojant programuojamas logines integralines mikroschemas ir bazinius matricos tipo kristalus.

Darbą su VHDL kalba galima suskaidyti į du komponentus: bendro algoritmo komponentą ir į klausimą orientuotą komponentą.

Bendro algoritmo komponentas – tai programinė kalba, kuri pagal sintaksę panaši į dabartines programavimo kalbas (tai Paskalis, C++ ir kitos kalbos). Ši programinė kalba priskiriama prie griežtai orientuotų kalbų klasės. Programiniame pakete yra integruoti (Standart tipo) paprasti (skaliarinio tipo) duomenys: loginio tipo aprašymai, duomenys aprašantys laiką ir kiti. Taip pat galima naudoti masyvus (Array), bitų vektorius (bit_vector) ir simbolines eilutes (string).

Nuosekliai vykdomi VHDL kalbos operatoriai gali būti panaudoti, aprašant procesus ir funkcijas. Juose yra:

1. Kintamųjų priskyrimo operatorius (:=)
2. Signalo priskyrimo nuoseklus operatorius (<=)
3. Tikrinimo nuoseklus operatorius (assert)
4. Sąlygos operatorius (IF)
5. Parinkimo operatorius (CASE)
6. Ciklo operatorius (LOOP)
7. Tuščias operatorius (NULL)
8. Procedūros-funkcijos grįžimo operatorius (RETURN)
9. Nuoseklaus procedūros iškvietimo operatorius (CALL)

Programinė kalba VHDL palaiko paketinio ir struktūrinio programavimo koncepcijas. Sudėtingi operatoriai yra apjungti operatoriuose: if- end if; process- end process; case- end case; loop- end loop ir panašiai.

Yra skirtumas tarp lokalių ir bendrų kintamųjų. Lokalūs kintamieji turi savo reikšmę tik bloko ribose.

Aprašymo fragmentai, kurie gali nepriklausomai būti analizuojami kompiliatoriaus, gali būti integruoti programiniame pakete MAX+plus II į darbo (Work) direktoriją. Ten jie bus pavadinti projekto paketais (design unit). Tokiais paketais gali būti projekto sąsajos aprašymas (entity), architektūros aprašymas (architecture), projekto konfigūracijos aprašymas (configuration), paketo sąsajos aprašymas (package) ir kūno paketo aprašymas (package body).

Projekto modulius galima irgi padalinti į dvi kategorijas: pirminius ir antrinius. Prie pirminių paketų priskirti paketo konfigūracijos aprašymai. Prie antrinių priskirti paketo architektūros ir paketo kūno aprašymai. Vienas arba keli projekto moduliai gali būti viename faile, kuris vadinsis projekto failu (Design file).

Kiekvienas išanalizuotas projekto modulis integruojamas į Design Library vietą ir bus skaitomas kaip modulis (Library unit). Kiekviena tokia vieta VHDL kalboje turi savo loginį vardą.

Atliekant projektavimo darbą su VHDL kalba, egzistuoja dviejų tipų projekto darbo bazės: paprastos darbo bazės ir resursų bazės. Paprasta darbo bazė – tai bazė Work, su kuria vartotojas dirba konkrečiame projekte ir į kurią integruoja paketą, kuris buvo sukurtas analizuojant projekto paketą. Resursų bazė – tai bazė, kurioje yra moduliai, prie kurių galimas priėjimas iš projekto analizuojančio modulio. Vartotojas visada dirba su viena paprasta darbo baze ir su norimu resursų bazių kiekiu.

Kaip ir paprastose algoritminėse kalbose, moduliai – tai išrinkti iš programų duomenų, kintamųjų, procedūrų ir funkcijų dalys, kurios duoda galimybę supaprastinti projektavimo darbą. Pavyzdžiui, tai supaprastina projekto dalių pakeitimą.

VHDL programinėje kalboje sąsajų aprašymas ir paketo kūno aprašymas yra skirtingas. Pagal nutylėjimą numatytas standartinių paketų Standart ir Text 10 prijungimas prie projekto. Pavyzdžiui, pakete Standart yra loginių operacijų aprašymai su bitų vektoriumi. Nestandartinių tipų paketai realizuojami norint su didesniu tikslumu atvaizduoti aprašymų objektų savybes. Pavyzdžiui, galima iš anksto pakeisti logines IR, ARBA ir NE operacijas ir dirbti ne su 0 ir 1 reikšmėmis, o su daugiareikšme (1, 0, X, Z ...) modeliavimo abėcėle.

Į klausimą orientuotas komponentas duoda galimybę aprašyti skaitmenines sistemas naudojant pažįstamus terminus. Prie jų gali būti priskirti:

- Modeliavimo laiko supratimas NOW.
- Time tipo duomenys, kurie nurodo užlaikymo laiką fizikiniais skaičiais.
- Signal tipo signaliniai duomenys, kurių būseną keičiama ne iškart, kaip pas paprastus kintamuosius, o su nurodytu užlaikymu. Taip pat prie jų priskiriamos specialios operacijos ir funkcijos su signaliniais duomenimis.
- Galimybės aprašyti objektus Entity tipo ir jų architektūros (architecture) tipą.

Kaip ir visos kitos programavimo kalbos, VHDL turi savo abėcėlę į kurią įeina simbolių rinkinys. Rinkinyje yra:

1. Lotynų didelės ir mažos raidės : A, B, . . . , Z ir a, b, . . . , z.
2. Skaičiai nuo 0 iki 9.
3. Pabraukimo simbolis „_“ (ASCII kodo numeris 95).

Tik iš šių simbolių gali būti konstruojamos programos dalys. Kitų simbolių panaudojimas kompiliatoriuje bus suprantamas kaip klaidingas užrašymas.

Programinėje kalboje VHDL komentaras parodomas panaudojant „--“ simbolį. Kompiliatorius ignoruoja tekstą, kuris prasideda nuo „--“ simbolio. Šiuo atveju komentaras gali turėti raidžių tipus, kurių nėra galimų raidžių sąrašė.

Programinės kalbos standartas gali dirbti su įvairių tipų skaičiais, bet PLM sintezavimo priemonės leidžia dirbti tik su sveikais skaičiais. Sveikas skaičius gali būti pavaizduotas vienoje iš skaičiavimo sistemų: dvejetainėje, aštuntainėje, dešimtainėje ir šešioliktainėje.

Simbolių užrašymas – tai simbolis, esantis kabutėse. Pavyzdžiui, ‘A’ arba ‘B’.

Programinėje kalboje yra daug paprastų ir sudėtingų duomenų tipų. Prie paprastų tipų priskiriami tokie duomenų tipai:

1. BOOLEAN (loginis tipas) – šio tipo objektai gali būti FALSE (melas) ir TRUE (tiesa) būsenos.
2. INTEGER (sveikas) – šio tipo objektai tai skaičiai iš $-(231-1)\dots 231-1$ (-2147483647 ... 2147483647) diapazono.
3. BIT (bitas) – tai vienas loginis bitas. Šio tipo objektai gali būti „0“ arba „1“.
4. STD_LOGIC (bitinis tipas) – tai vienas duomenų bitas. Šio tipo objektai gali būti devynių būsenų. Šis tipas aprašytas pagal IEEE 1164 standartą.
5. STD_ULOGIC (bitinis tipas) – tai vienas duomenų bitas. Šio tipo objektai gali būti devynių būsenų. Šis tipas irgi aprašytas pagal IEEE 1164 standartą.
6. ENUMERATED (numeruojamas tipas) – skirtas užduoti vartotųjų tipus.
7. SEVERITY_LEVEL – numeruojamas tipas, kuris naudojamas tik ASSERT operatorių.
8. CHARACTER – simbolinis tipas.

Iš visų sudėtingų tipų loginių schemų programavime naudojami tik masyvai (ARRAY) ir užrašymai (RECORD tipas). Pavyzdžiui, šių masyvų tipai yra iš anksto numatyti:

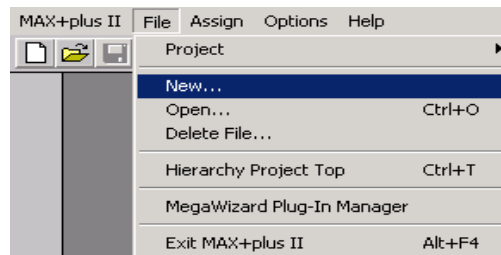
1. BIT_VECTOR – BIT tipo elementų masyvas.
2. STD_LOGIC_VECTOR – STD_LOGIC elementų masyvas.
3. STD_ULOGIC_VECTOR – STD_ULOGIC tipo elementų masyvas.
4. STRING – CHARACTER tipo elementų masyvas.

Masyvų indeksai ir ribos turi būti nurodyti tiesiogiai aprašant duotus objektų tipus.

5.3 Tipinių kombinacinių grandinių modeliavimas

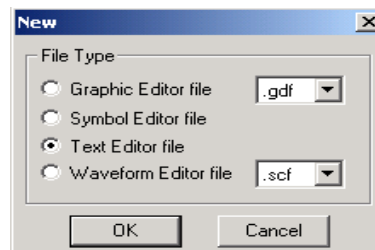
5.3.1 Šifratoriaus modeliavimas tekstiniu būdu

Šifratorius (CD) – tai tipinė skaitmeninė kombinacinė grandinė, kuri signalą viename iš n įėjimų keičia į dvejetainį m skilčių kodą su vertėmis 8-4-2-1 išėjime. Norint atlikti šifratoriaus modeliavimą su Max+plus II programiniu paketu, reikia File lentelėje paspausti New.



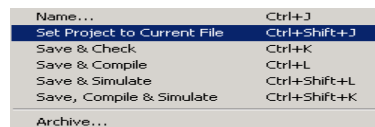
5.1 pav. Naujo falų sudarymas programiniame pakete Max+plus II

Paspaudus New atsiranda galimybė išrinkti kūrimo objekto modeliavimo būdą. Modeliuojant šifratorių, bus pasirinktas sukūrimas jo tekstiniame režime – Text Editor File.



5.2 pav. Sudaromo objekto projektavimo režimo pasirinkimas

Sukūrimo objekto patikrinimas ir kompiliavimas gali būti padarytas tik tada, kai objektui bus atlikta operacija Set Project to Current File.



5.3 pav. Priskyrimas sukurto objekto darbiniam projektui

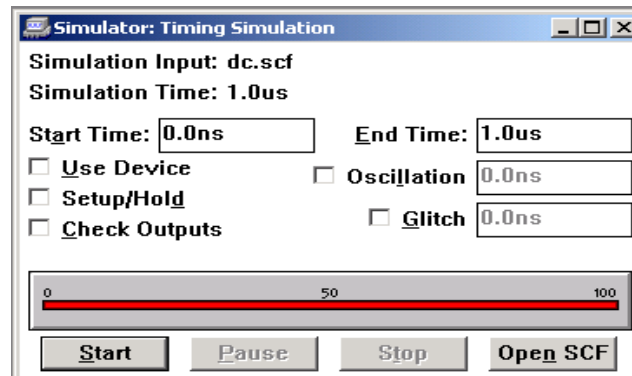
Sukurta programa toliau atrodo taip:

```

SUBDESIGN CD
(
IN[8..1]          :INPUT;
Binary_code[3..1] :OUTPUT;
)
BEGIN
  TABLE
IN[] => Binary_code[];
  b"00000001" => 0;
  b"00000010" => 1;
  b"00000100" => 2;
  b"00001000" => 3;
  b"00010000" => 4;
  b"00100000" => 5;
  b"01000000" => 6;
  b"10000000" => 7;
  END TABLE;
END;

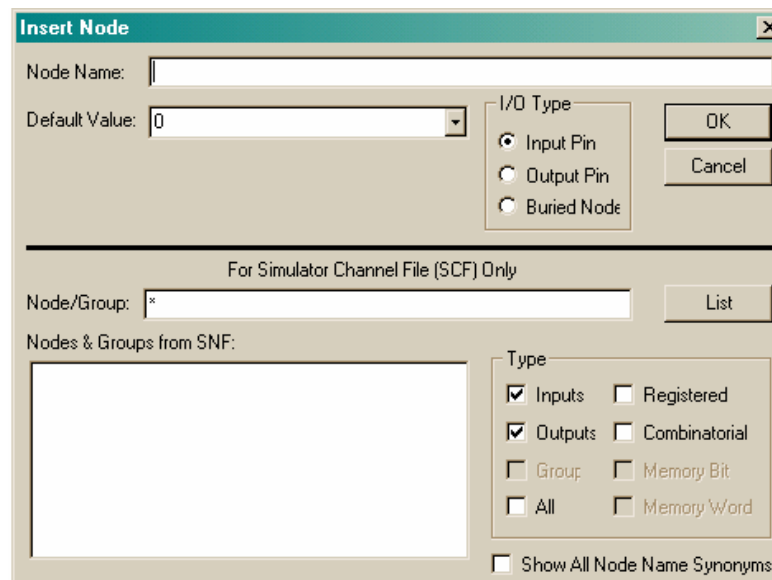
```

Norint simuliuoti programą, išreikštą tekstu, reikia File lentelėje paspausti Simulator. Atidarytame langelyje, kuris pavaizduotas 8.4 pav., galima nustatyti modeliavimo laiko pradžią ir pabaigą, naudojamos mikroschemos automatinį patikrinimą.

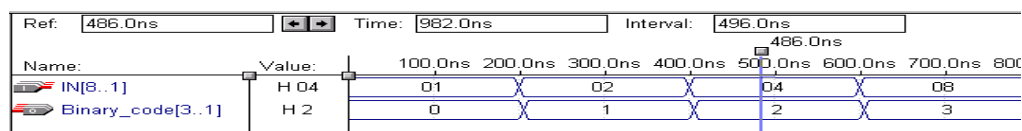


5.4 pav. Simuliator funkcijos parametrų nustatymo lentelė

Pagal sukurtą programą galima atlikti programos grafinį testavimą. Norint tai atlikti, reikia nurodyti kokie įėjimai ir kokie išėjimai yra projekte. Taip pat reikia nurodyti modeliavimo laiko trukmę. Tai daroma paspaudus .scf failą.

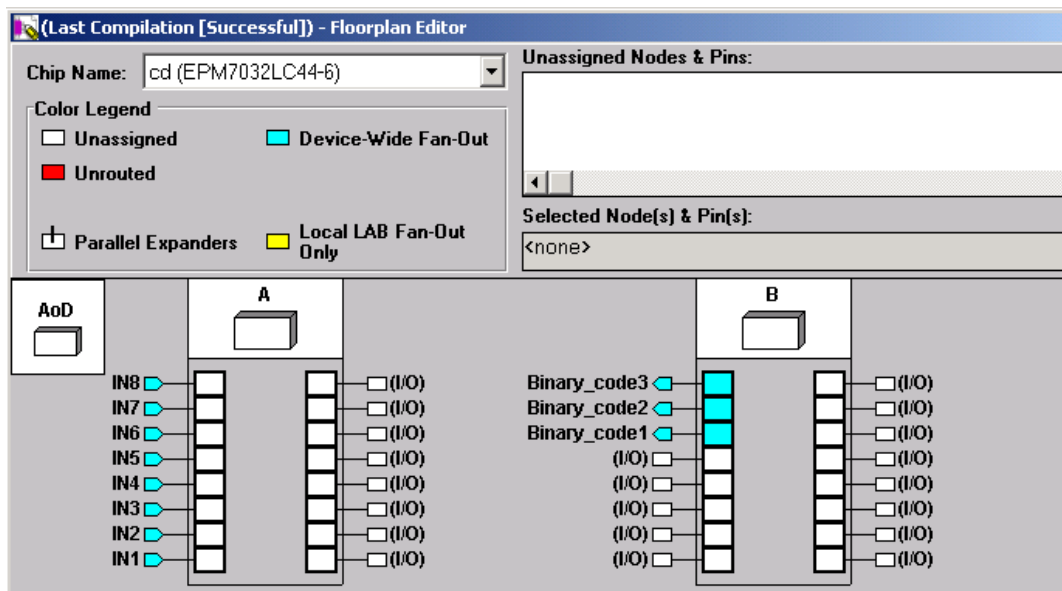


5.5 pav. Įėjimų ir išėjimų instaliavimas projekte



5.6 pav. Šifratoriaus modeliavimo rezultatai grafiniame režime

Ateinantiems signalams galima priskirti norimą būseną ir nurodyti norimus laiko intervalus. Po simulate mygtuko paspaudimo kombinacinės grandinės simulatorius nurodys išėjimo signalo būsenas.



5.7 pav. Floorplan Editor funkcijos panaudojimas šifratoriaus modeliavime

Floorplan Editor leidžia pamatyti sukurtos schemos siūlomą kojelių PLM išdėstymą. Taip pat galima pasirinkti PLM tipą. PLM tipas EPM7032LC44-6 yra integruotas programiniame pakete, todėl visi kiti modeliavimo rezultatai demonstruojami su šiuo PLM tipu.

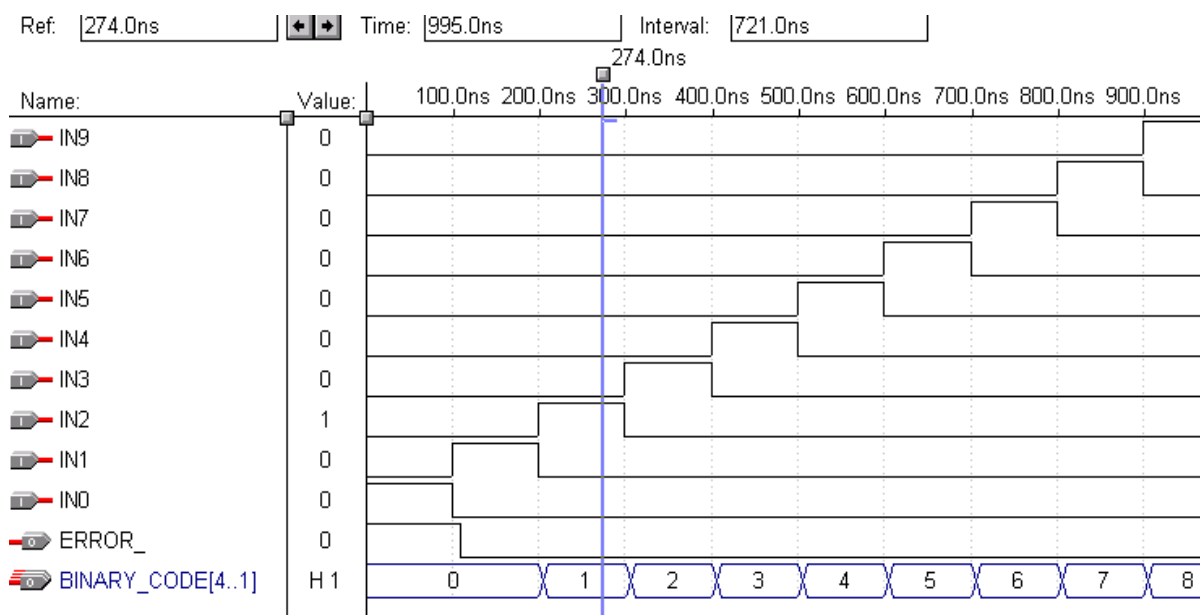
Tekstiniame apraše pavaizduota programa yra pati paprasčiausia. Joje nepanaudotas operatorius Defaults ir visoms nenurodytoms įėjimo reikšmėms išėjimo signalams bus priskirtas B"000" lygis. Duomenų įvedimui klaviatūros pagalba dažniausiai naudojamas nepilnas šifratorius 10□4. Tokiame šifratoriuje kiekvienam klaviatūros dešimtainiam skaičiui priskiriamas dvejetainis kodas. Galima pamatyti atsiradusias klaidas, jei paspausim daugiau negu vieną mygtuką arba nepaspausim nei vieno.

Tokiu atveju programa atrodys taip:

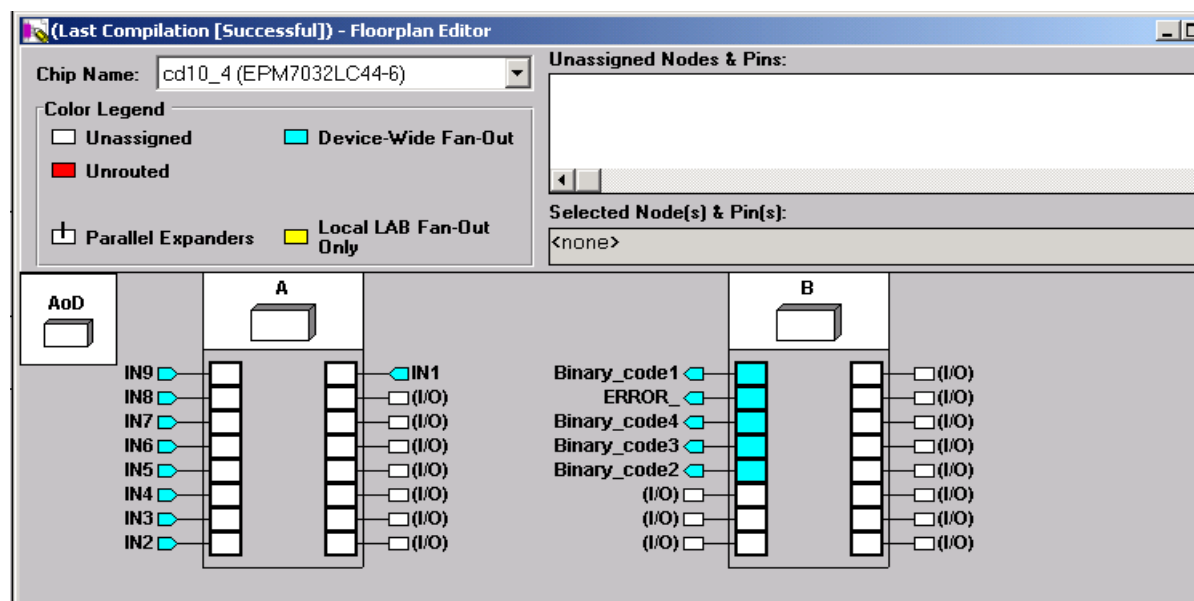
```
SUBDESIGN CD10_4
(
IN[9..1]           :INPUT;
Binary_code[4..1], ERROR_   :OUTPUT;
)
BEGIN
CASE IN[] IS
WHEN H"001" => Binary_code[] = B"0000";
WHEN H"002" => Binary_code[] = B"0001";
WHEN H"004" => Binary_code[] = B"0010";
WHEN H"008" => Binary_code[] = B"0011";
WHEN H"010" => Binary_code[] = B"0100";
WHEN H"020" => Binary_code[] = B"0101";
WHEN H"040" => Binary_code[] = B"0110";
WHEN H"080" => Binary_code[] = B"0111";
WHEN H"100" => Binary_code[] = B"1000";
WHEN OTHERS => ERROR_ = VCC;
END CASE;
END;
```


Šioje programoje Binary_code[4..1] ir ERROR_ loginės funkcijos reikšmės užduotos ne visiems funkcijos galimiems argumentams, todėl, kaip ir pirmu šifratoriaus modeliavimo atveju, jie turės loginio nulio būseną.

Pagal sukurtą programą galima atlikti modeliavimą grafiniu režimu užduodant įėjimo signalo reikšmes. Šiuo atveju įėjimo signalams galima priskirti loginio nulio arba vieneto būseną. Įėjimo signalo reikšmės buvo priskirtos vienos po kitos, norint modeliavimo rezultatus padaryti kuo aiškesnius. Taip pat kaip ir pirmu atveju mėlynos linijos nustatymas rodo Value lentelėje sumodeliuotos funkcijos būseną kiekvienu laiko momentu.



5.8 pav. Šifratoriaus 10 į 4 modeliavimo rezultatai pavaizduoti grafiškai

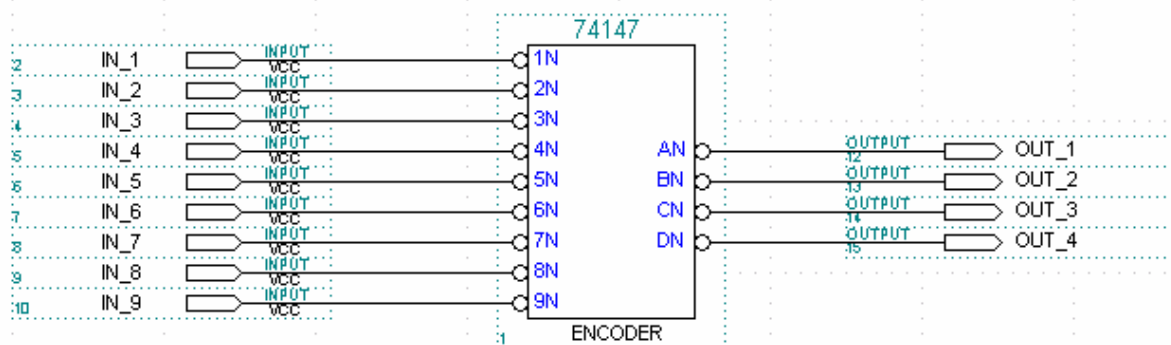


5.9 pav. Floorplan Editor funkcijos šifratoriaus 10 į 4 modeliavime panaudojimas

Kaip matome, antru šifratoriaus modeliavimo atveju mikroschemų kojelių išdėstymas bus kitoks dėl to, kad padidėjo išėjimų skaičius ir atsirado ERROR_ variantas.

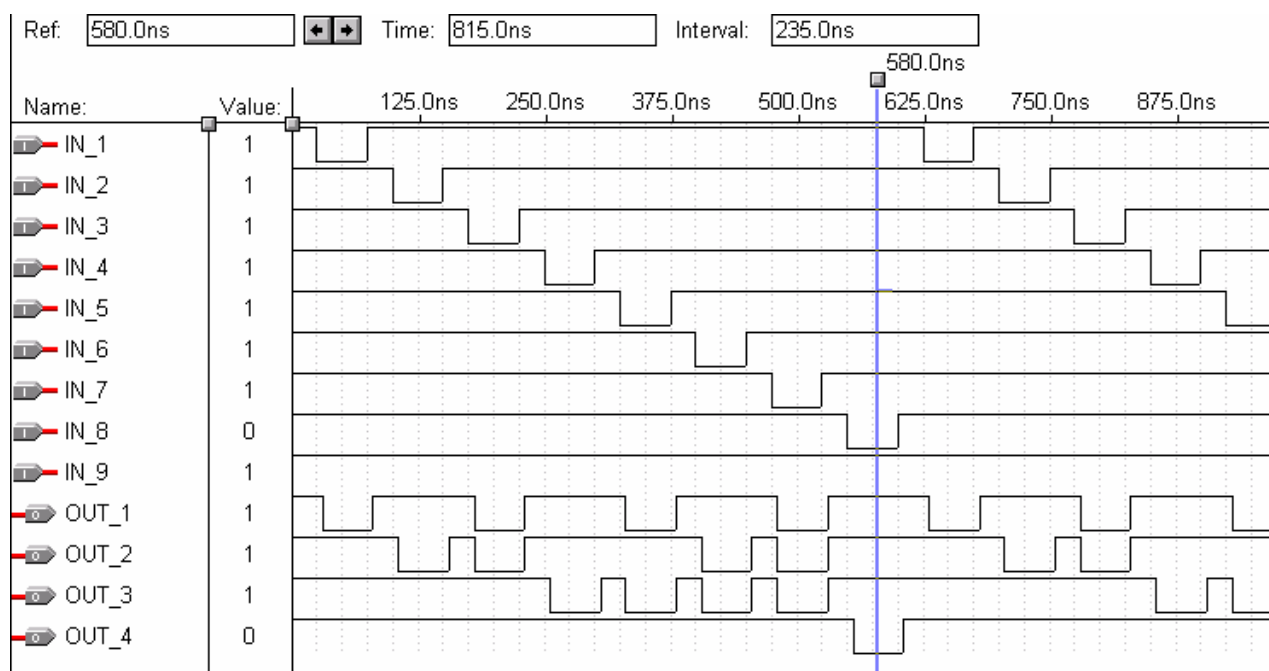
5.3.2 Šifrotoriaus modeliavimas grafiniu būdu

Šifrotoriaus modeliavimas grafiniu būdu pradedamas parenkant lentelėje File mygtuką New. Po to parenkamas grafinio modeliavimo būdas (Graphic Editor File). Projektui sudaroma operacija Set Project to Current File. Šifrotoriaus modeliavimas grafiniu būdu buvo atliktas panaudojant integruotą į elementų bazę 74147 tipo mikroschemą. Šio šifrotoriaus mikroschema dažniausiai naudojama dešimtainės informacijos keitimui į BCD kodą.

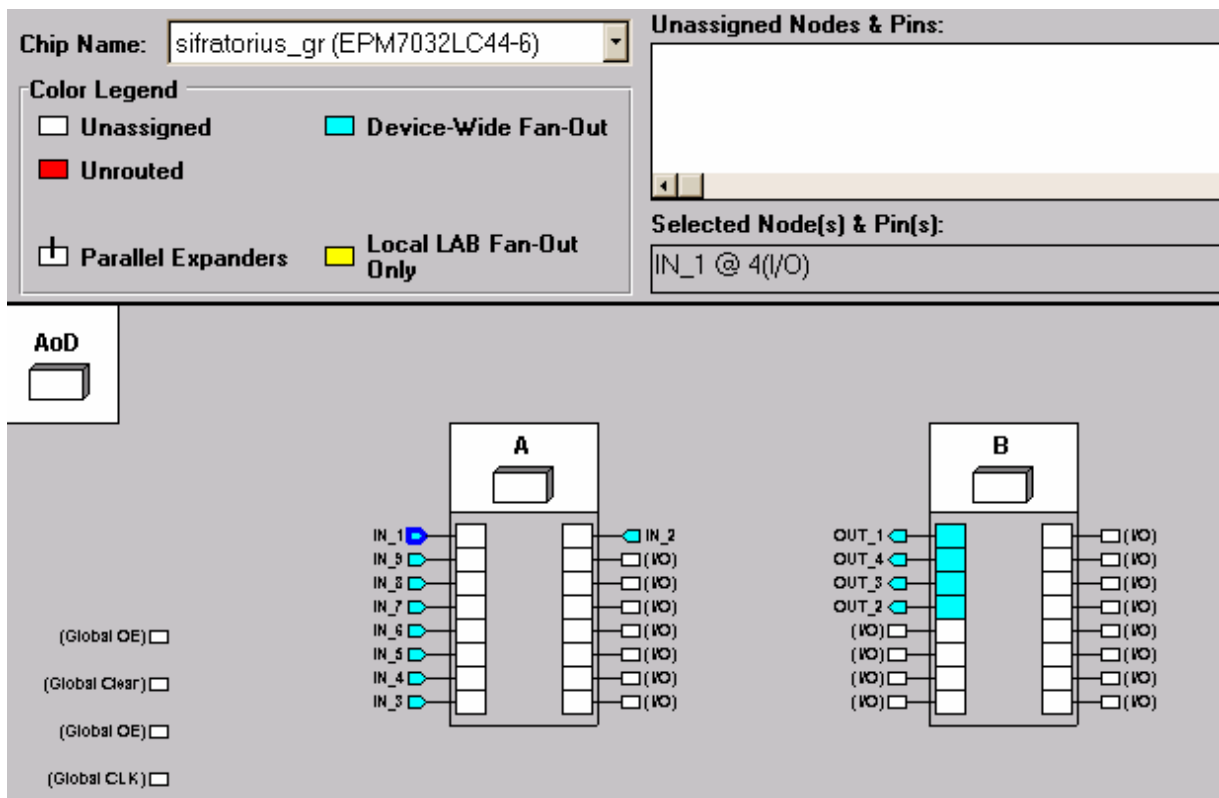


5.10 pav. Šifrotoriaus „iš 10 į BCD“ modeliavimas grafiniu būdu

Dėl to, kad šiame mikroschemos modelyje yra invertuoti įėjimai ir išėjimai, modeliavimo rezultatai atrodo taip:



5.11 pav. Šifrotoriaus „iš 10 į BCD“ grafiniu būdu modeliavimo rezultatai



5.12 pav. Floorplan Editor funkcijos šifratoriaus „iš 10 į BCD“ modeliavime panaudojimas

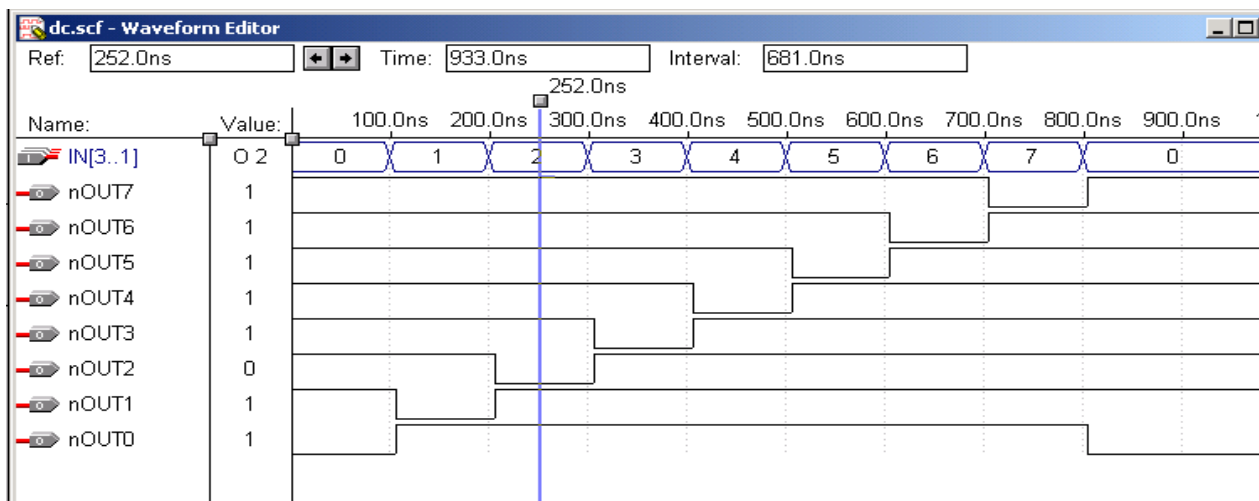
5.3.3 Dešifratorius modeliavimas tekstiniu būdu

Dešifratorius (DC) – tai tipinė skaitmeninė kombinacinė grandinė, kuri dvejetainį kodą su vertėmis 8-4-2-1 įėjime keičia į reikiamo formato kodą išėjime. Pilnas dešifratorius turi n įėjimų ir 2^n išėjimų, nepilnas – mažiau negu 2^n išėjimų.

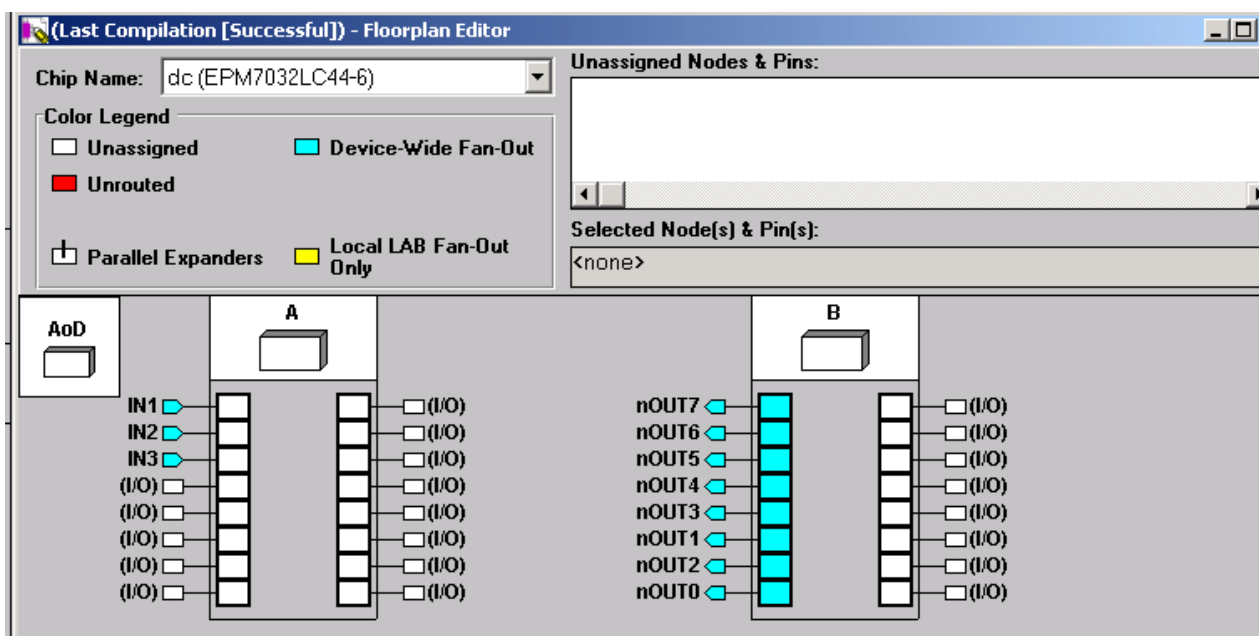
Norint atlikti dešifratoriaus modeliavimą su Max+plus II programiniu paketu reikia atlikti tokius pat veiksmus objekto sukūrimui.

Trijų skilčių pilno dešifratoriaus su inversiniais išėjimais projektavimas bus toks:
SUBDESIGN DC

```
(
IN[3..1]      :INPUT;
OUT[7..0]     :OUTPUT;
)
BEGIN
CASE IN[] IS
  WHEN 0 => nOUT[] = B"11111110";
  WHEN 1 => nOUT[] = B"11111101";
  WHEN 2 => nOUT[] = B"11111011";
  WHEN 3 => nOUT[] = B"11110111";
  WHEN 4 => nOUT[] = B"11101111";
  WHEN 5 => nOUT[] = B"11011111";
  WHEN 6 => nOUT[] = B"10111111";
  WHEN 7 => nOUT[] = B"01111111";
END CASE;
END;
```



5.13 pav. Dešifrotoriaus modeliavimo rezultatai pavaizduoti grafiškai

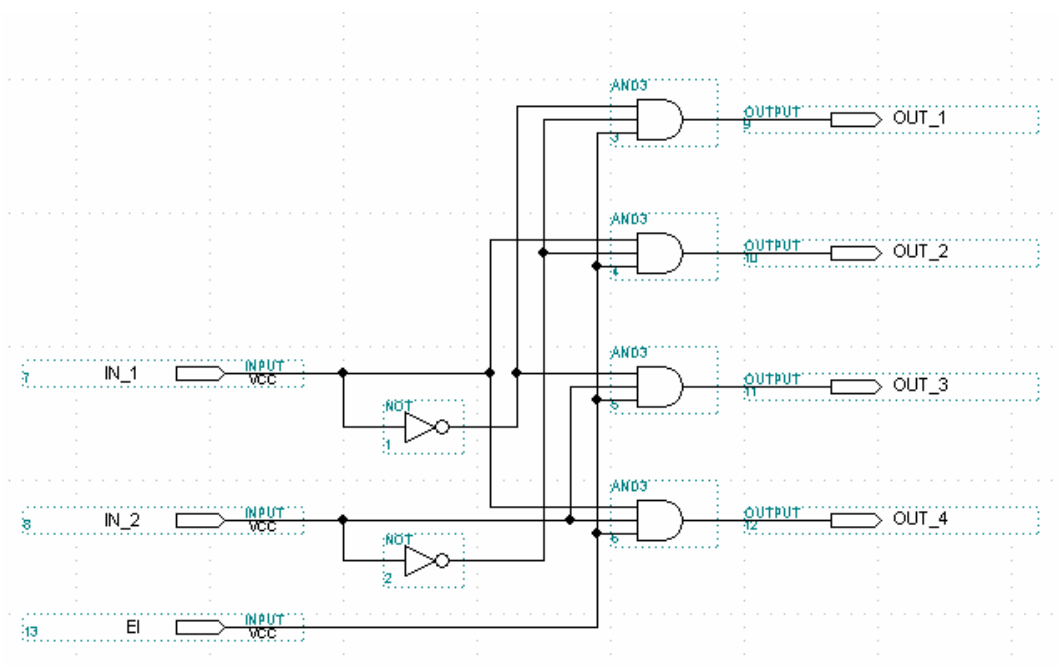


5.14 pav. Floorplan Editor funkcijos panaudojimas dešifrotoriaus modeliavime

5.3.4 Dešifrotoriaus modeliavimas grafiniu būdu

Dešifrotoriaus modeliavimas grafiniu būdu taip pat pradedamas parenkant lentelėje File mygtuką New. Tik dabar parenkamas grafinio modeliavimo būdas (Graphic Editor File). Projektui taip pat atliekama operacija Set Project to Current File.

Sukurtas projektas grafiškai atrodo taip:



5.15 pav. Dešifраторius „iš 2 į 4“ projektavimas

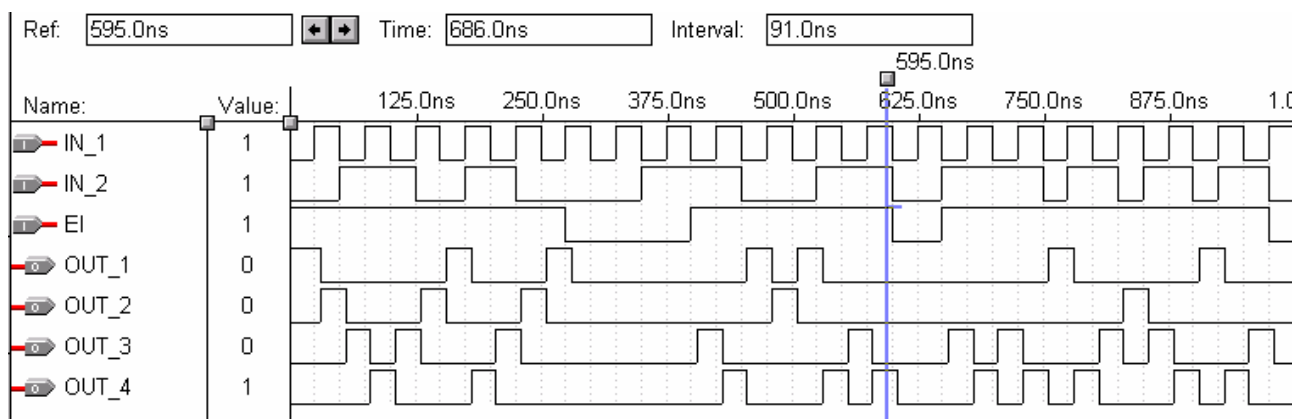
Pavaizduotas dešifраторius turi įėjimo leidimo įėjimą EI, du tiesioginius duomenų įėjimus, į kuriuos paduodamas dviskiltis dvejetainis kodas ir keturis tiesioginius duomenų išėjimus. Informacija iš dešifраторiaus bus gaunama tik tuomet, kai EI=1. Jei EI=0, tai visuose išėjimuose bus 0, nepriklausomai nuo loginių lygių, esančių duomenų įėjimuose.

Sukurtas dešifраторius turi tokią reikšmių lentelę:

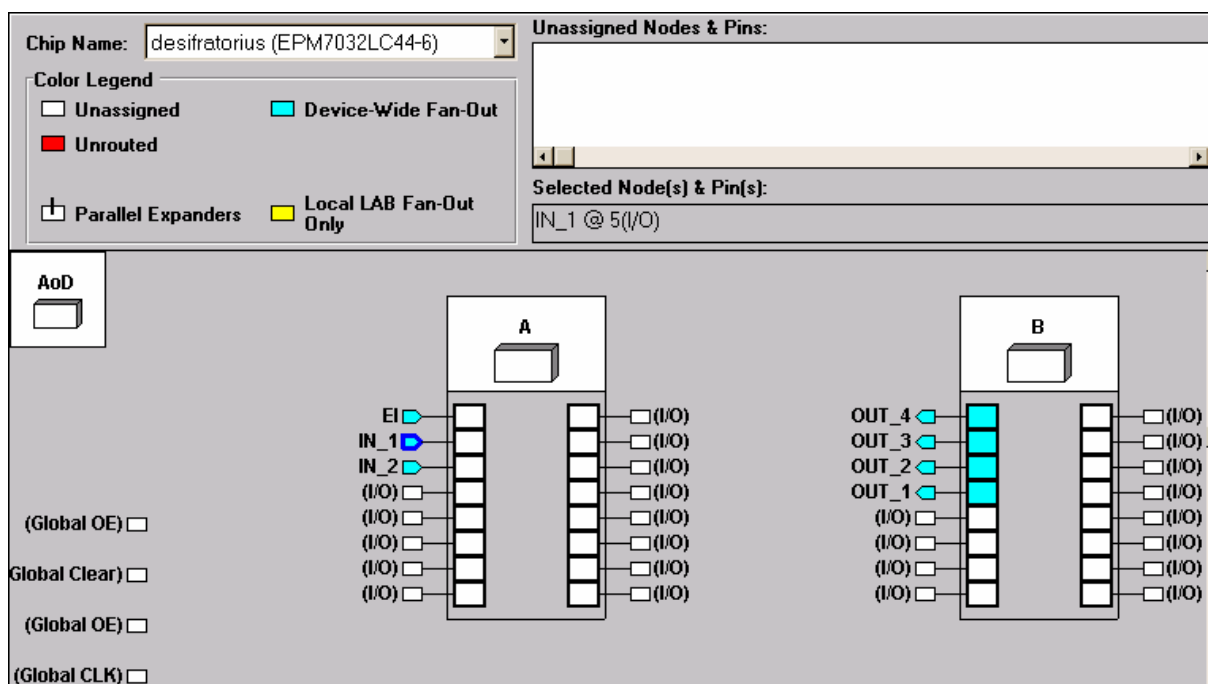
5.1 lentelė. Šifраторius „iš 2 į 4“ reikšmių lentelė

Įėjimai			Išėjimai			
EI	IN_2	IN_1	OUT_4	OUT_3	OUT_2	OUT_1
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	0	1	1	1

Modeliavimo rezultatai grafiniu būdu taip pat gaunami parenkant .scf. formato faile įėjimo signalus ir paspaudus mygtuką Simuliate:



5.16 pav. Dešifраторius „iš 2 į 4“ modeliavimo rezultatai



5.17 pav. Floorplan Editor funkcijos panaudojimas dešifраторius „iš 2 į 4“ modeliavime

5.3.5 Multiplexerio modeliavimas tekstiniu būdu

Multiplexeris (MUX) – tai kombinacinė schema, kuri leidžia nuosekliai apklausti daugumą įėjimų ir perdavinėti jų būsenas per vieną išėjimą. Multiplexeriuose informacinių įėjimų skaičius dažniausiai lygus $2n$, kur n – adresinių įėjimų skaičius.

Multiplexerio modeliavimo etapai buvo parinkti tokie pat, kaip ir kitų kombinacinių grandinių.

Tekstiniame režime multiplexerius, kuris turi keturis informacinius įėjimus (IN[4..1]), du adresinius įėjimus (ADR[2..1]) ir vykdyti darbo galimybė (ENABLE) turi tokią tekstinę formą:

```
SUBDESIGN MUX
```

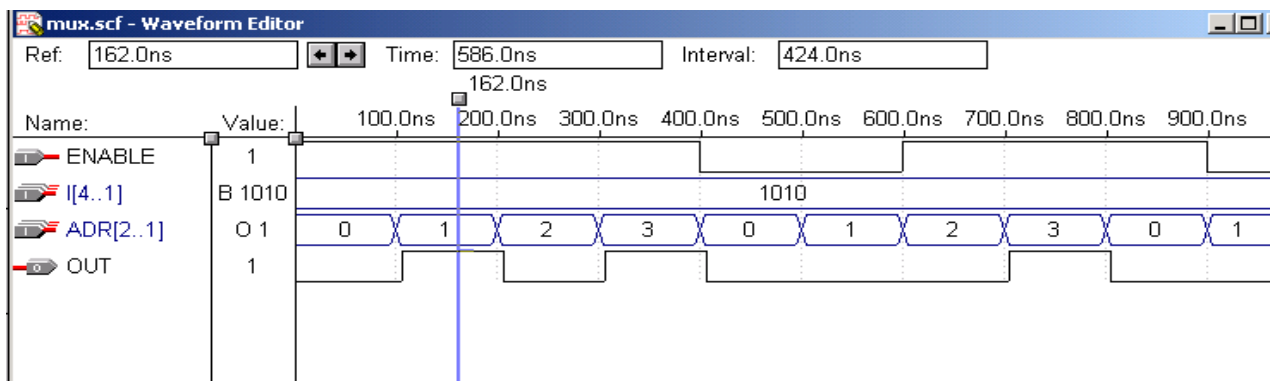
```
(
  I[4..1], ADR[2..1], ENABLE : INPUT;
  OUT : OUTPUT;
```

```

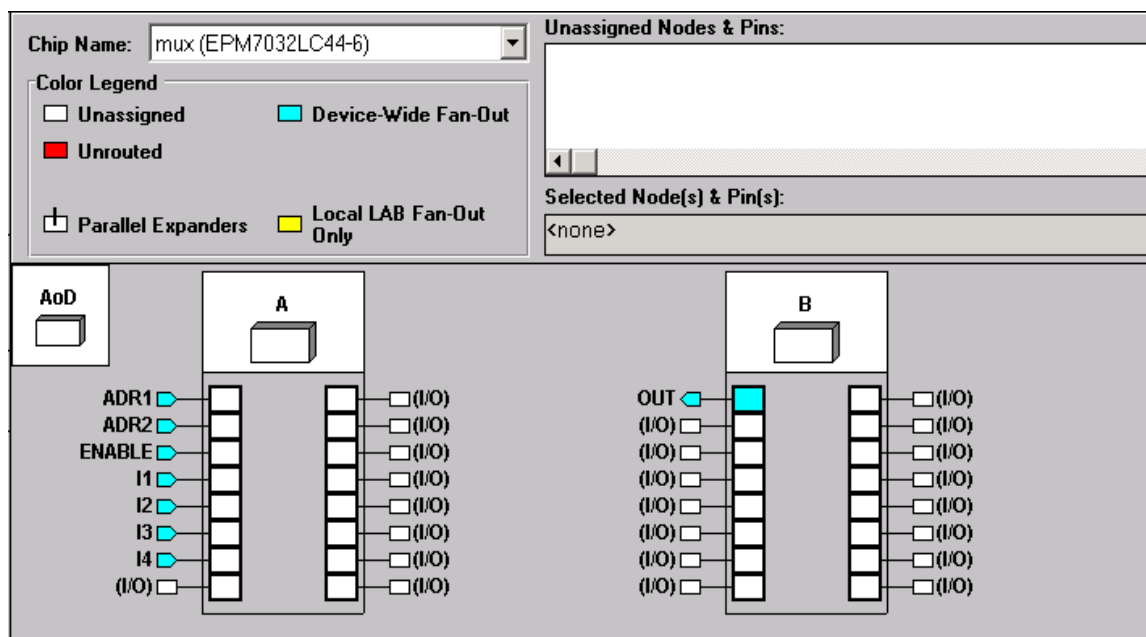
)
BEGIN
  IF ENABLE THEN
    CASE ADR[] IS
      WHEN 0 => OUT = I[1];
      WHEN 1 => OUT = I[2];
      WHEN 2 => OUT = I[3];
      WHEN 3 => OUT = I[4];
    END CASE;
  END IF;
END;

```

Jeigu ENABLE = 0, tai multiplexerio išėjimuose bus loginis nulis nepriklausomai nuo kitų įėjimų būsenos (nes šiame tekstiniame aprašyme loginė funkcija OUT neįvertina visų galimų variantų). Jeigu ENABLE = 1, tai multiplexerio išėjimo signalas bus apsprendžiamas pagal adresinį kodą ADR[] ir informacinio signalo lygį. Šioje sąlygoje sukurto multiplexerines kombinacinės grandinės modeliavimo rezultatai atrodys taip:



5.18 pav. Multiplexerio modeliavimo rezultatai grafiniame režime

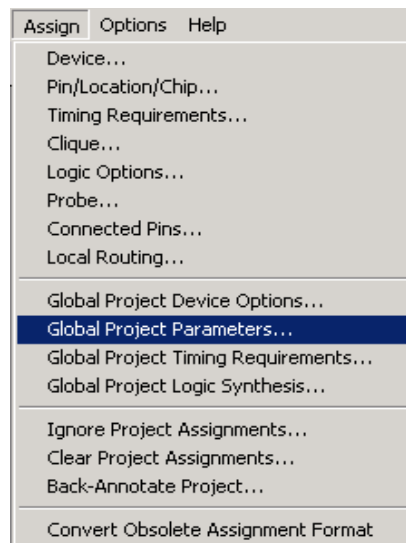


5.19 pav. Floorplan Editor funkcijos panaudojimas multiplexerio modeliavime

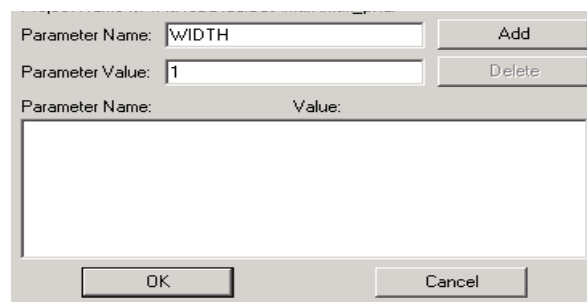
Norint sukurti parametruotą multiplekserį, kurio įėjime yra dvi WIDTH magistralės A[WIDTH..1] ir B[WIDTH..1] ir vienas adresas, reikia panaudoti ASSERT operatorių, kuris skirtas parametro WIDTH patikrinimui. Programa tokiu atveju bus tokia:

```
PARAMETERS (WIDTH= 8);
ASSERT (WIDTH != 1)
REPORT "Value of parametr WIDTH = %"WIDTH
SEVERITY INFO;
ASSERT (WIDTH > 0)
REPORT "Value of parametr WIDTH must be greater than %"WIDTH
SEVERITY INFO;
SUBDESIGN MUX_P
(A[WIDTH..1], B[WIDTH..1], ADR :INPUT;
OUT[WIDTH..1] :OUTPUT;)
BEGIN
  IF ADR
    THEN OUT[] = A[];
    ELSE OUT[] = B[];
  END IF;
END;
```

Programa nedirbs, jei parametru WIDTH nebus priskirta tam tikra reikšmė. Norint tai padaryti reikės Assign lentelėje parinkti Global Project Parameters komandą:



5.20 pav. Global Project Parameters komandos Assign lentelėje parinkimas



5.21 pav. Parametru WIDTH vieneto priskyrimas

Pavyzdžiui panaudojant komandą Global Project Parameters kuri yra Assign lentelėje WIDTH priskirti vienetai, tai modulio kompiliavimo metu bus suformuotas pranešimas, kuris informuos apie parametro priskyrimą:

Info: Line 2, File f:\Mokslas\mux\mux_p.tdf:

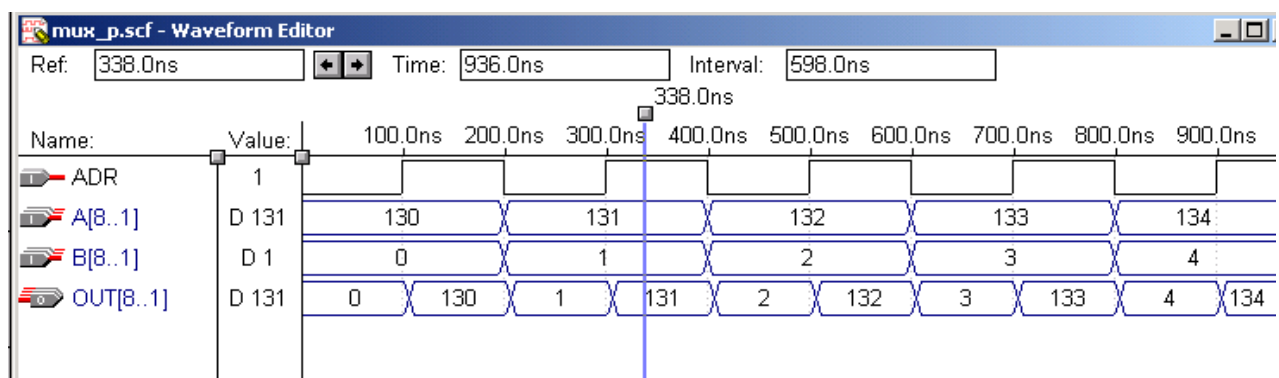
Value of parametr WIDTH=1.

„Griežtumo riba“ pranešimui apsprendžia INFO, todėl kompiliatorius pratęs darba. Bet jeigu parametrui WIDTH bus priskirtas nulis, bus suformuotas pranešimas:

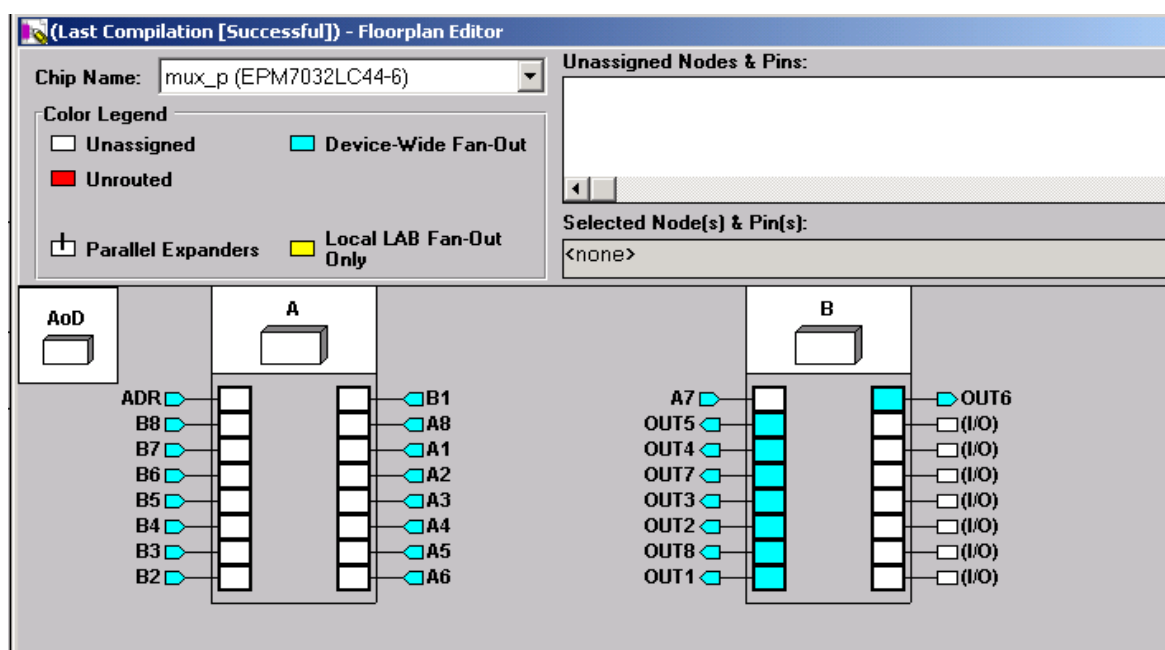
Error: Line 5, File f:\Mokslas\mux\mux_p.tdf:

Value of parametr WIDTH must be greater than 0.

Multiplexerio simuliacijos rezultatai grafiniame režime bus tokie:



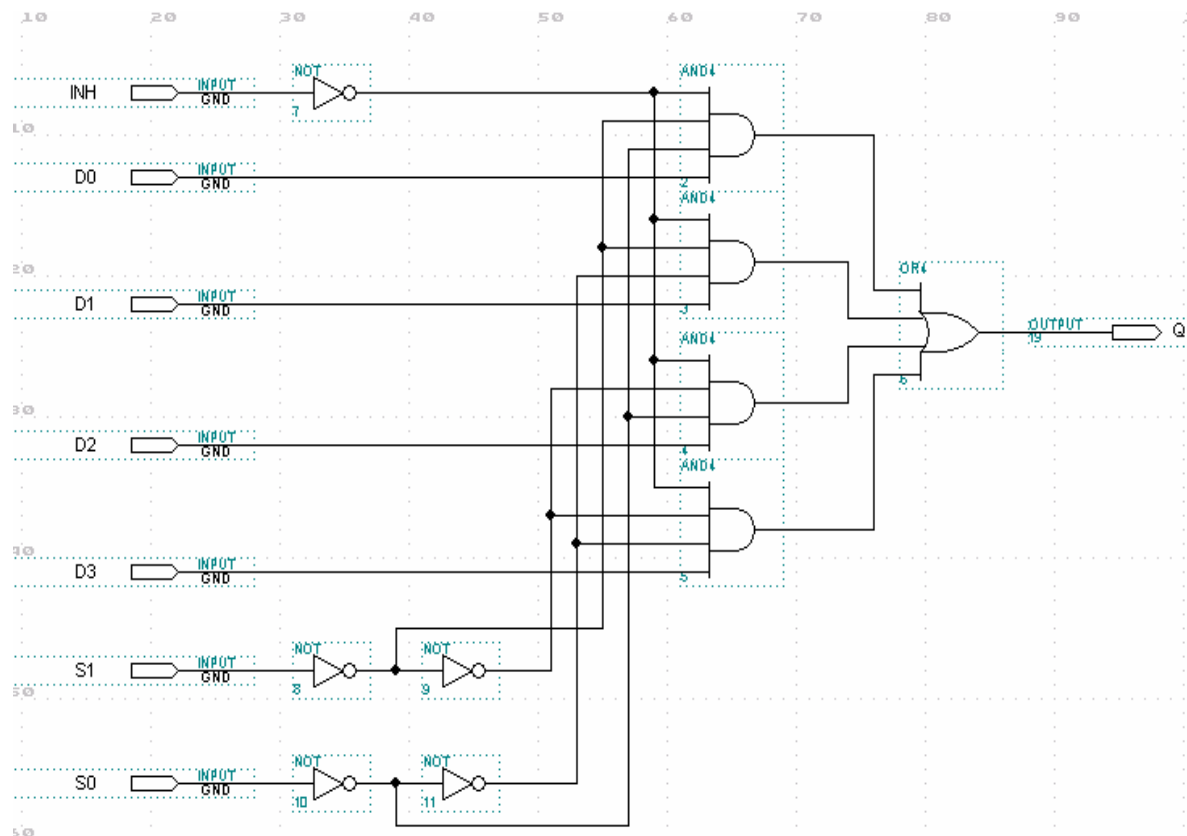
5.22 pav. Parametrizuoto multiplexerio modeliavimo rezultatai pavaizduoti grafiškai



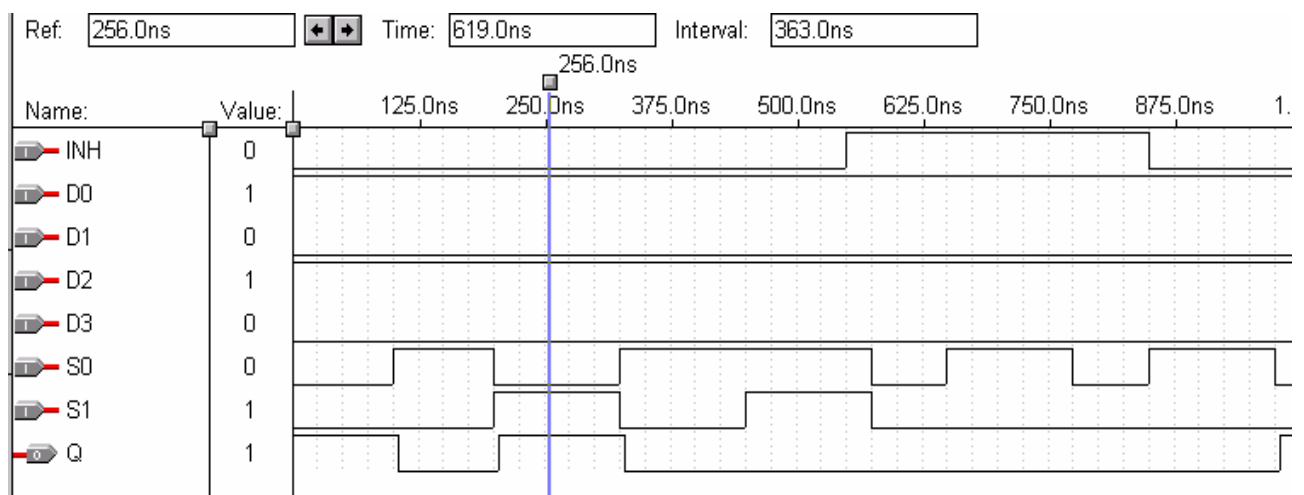
5.23 pav. Floorplan Editor funkcijos panaudojimas parametrizuoto multiplexerio modeliavime

5.3.6 Multiplexerio modeliavimas grafiniu būdu

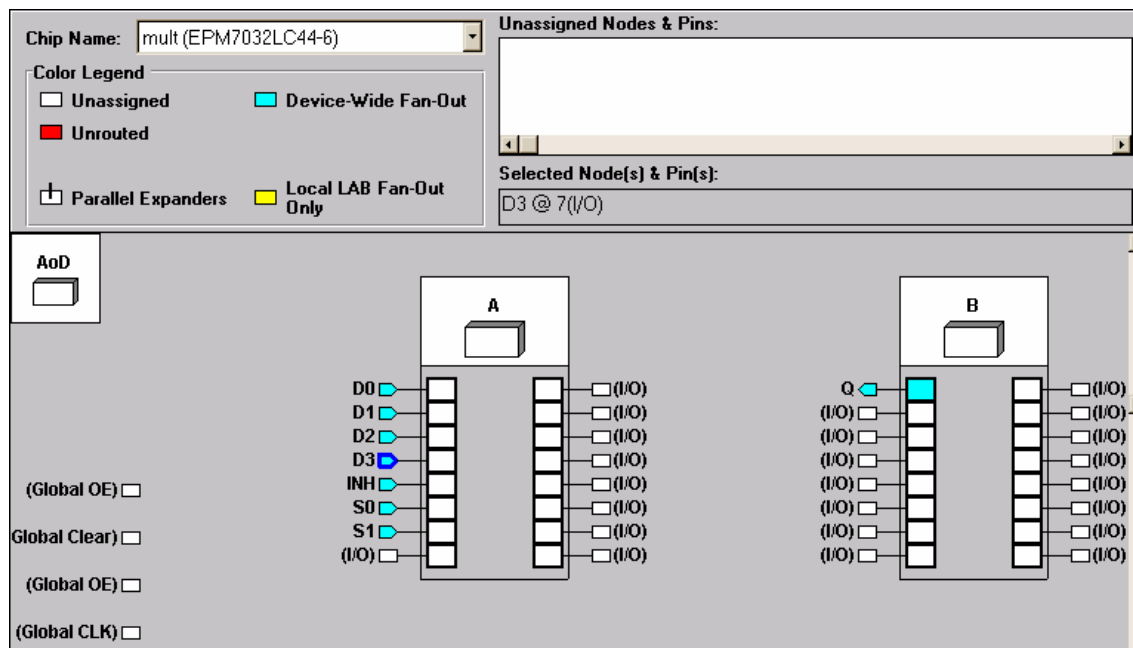
Multiplexerio projektavimas grafiniu būdu pradedamas nuo loginės schemos piešimo. Multiplexeris „iš 2 į 4“ su INH įėjimu (invertuotas įėjimo leidimo įėjimas) sudaromas iš keturių elementų IR, vieno ARBA ir penkių invertorių.



5.24 pav. Multiplexerio „nuo 4 prie 1“ loginė schema



8.25 pav. Multiplexerio „nuo 4 prie 1“ modeliavimo rezultatai



5.26 pav. Floorplan Editor funkcijos panaudojimas multiplekserio modeliavime

5.3.7 Demultiplekserio modeliavimas tekstiniu būdu

Demultiplekseris (DMX) – tai skaitmeninis kombinacinis įtaisas, skirtas informacijai, ateinančiai viena įėjimo linija, perduoti į vieną parinktą išėjimą. Išėjimo linija parenkama pagal adreso kodą. Jei n yra adreso ilgis, tai bus 2^n adresų kombinacijų ir tiek pat išėjimų linijų. Pagal parinktą adreso dvejetainį kodą, įėjimo duomenys perduodami į parinktą išėjimą.

Demultiplekseris, kuris turi vieną informacinį įėjimą IN, trijų skilčių adreso ilgį $ADR[3..1]$ ir aštuonis išėjimus $OUT[7..0]$ atrodo taip:

```
SUBDESIGN DMX
```

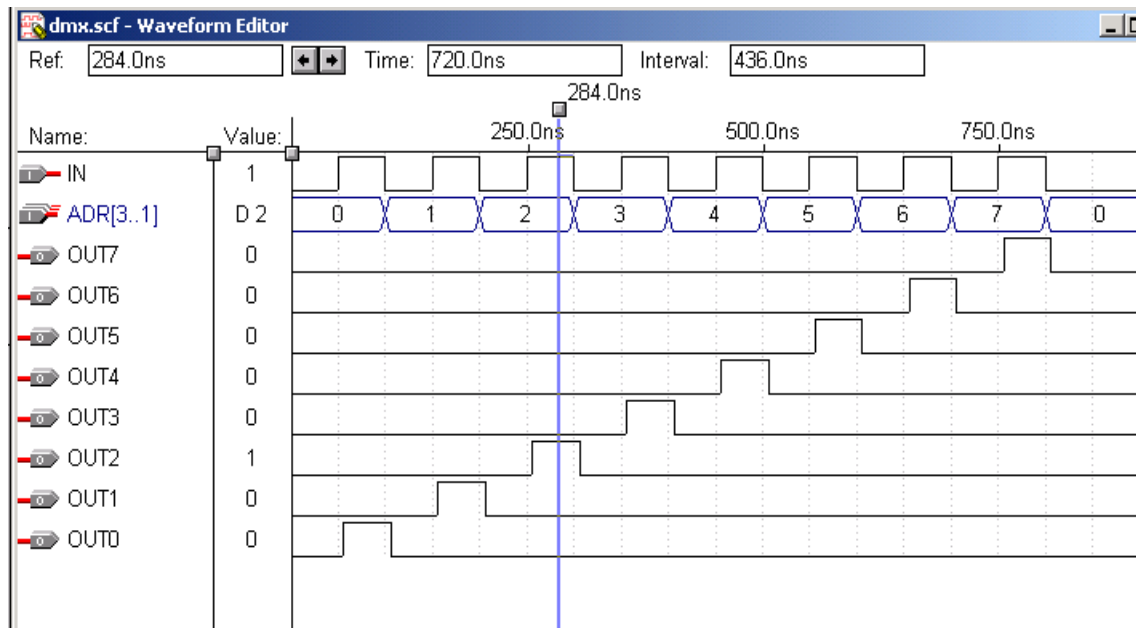
```
(
  IN, ADR[3..1] :INPUT;
  OUT[7..0] :OUTPUT;
)
```

```
BEGIN
```

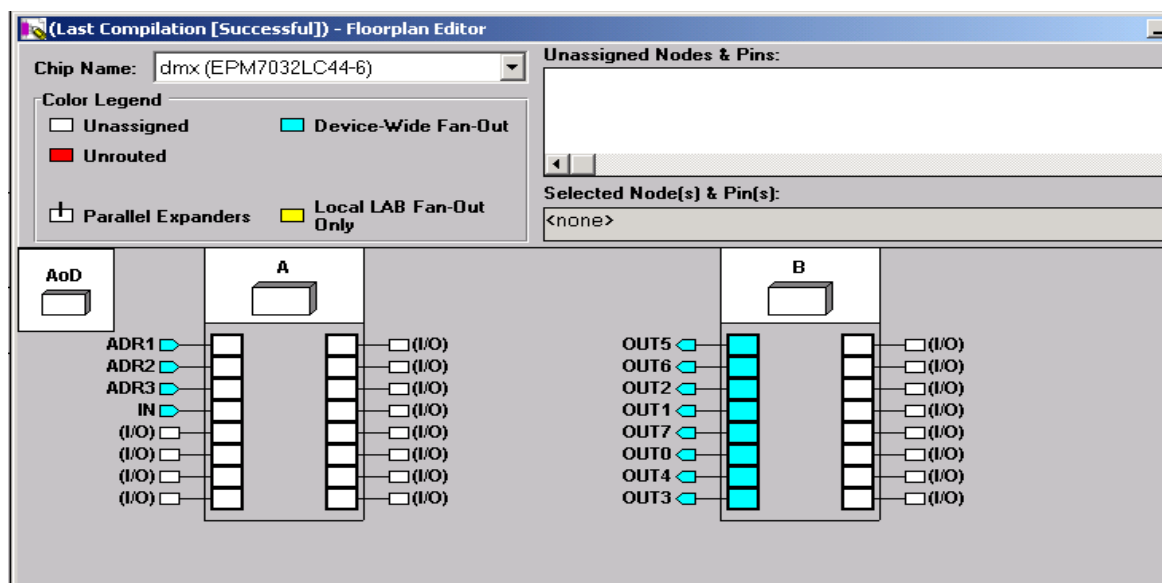
```
  CASE ADR[] IS
    WHEN 0 => OUT[0] = IN;
    WHEN 1 => OUT[1] = IN;
    WHEN 2 => OUT[2] = IN;
    WHEN 3 => OUT[3] = IN;
    WHEN 4 => OUT[4] = IN;
    WHEN 5 => OUT[5] = IN;
    WHEN 6 => OUT[6] = IN;
    WHEN 7 => OUT[7] = IN;
  END CASE;
```

```
END;
```

Kiekvienoje eilutėje CASE operatoriaus užduodama reikšmė tik vienam modulio išėjimui. Šioje programoje operatorius Defaults nebuvo panaudotas, todėl visuose kituose išėjimuose bus loginiai nuliai. Modeliavimo rezultatai atrodys taip:



5.27 pav. Demultiplekserio modeliavimo rezultatai grafiniame režime



5.28 pav. Floorplan Editor funkcijos panaudojimas demultiplekserio modeliavime

Demultiplekseris, kuris turi WIDTH skilčių informacinį įėjimą (IN[]), trijų skilčių adreso įėjimą (ADR[3..1]) ir aštuonis WIDTH skilčių išėjimus (OUT[7..0] [WIDTH..1]) atrodo taip:

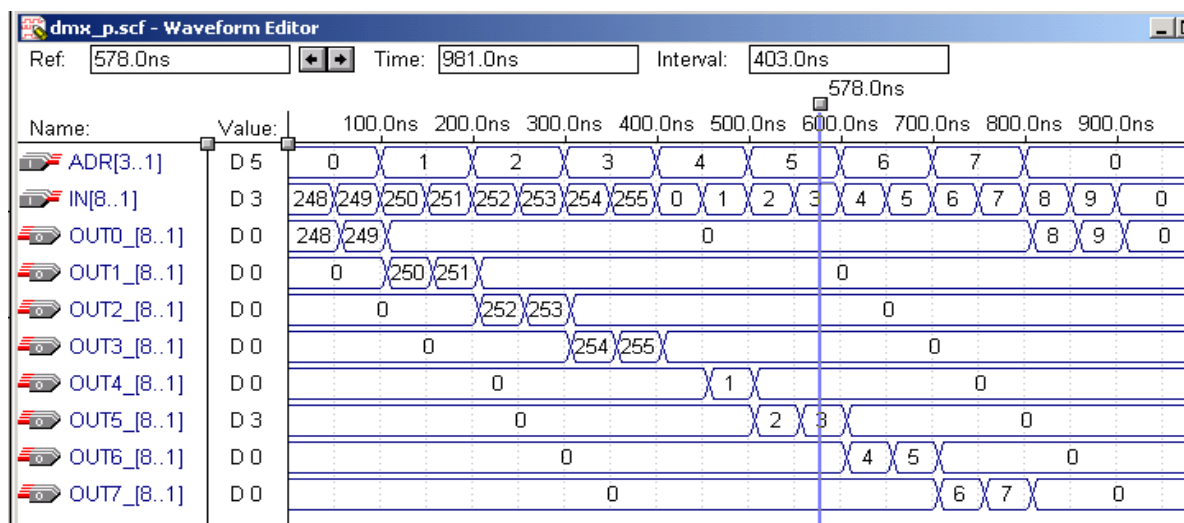
```
PARAMETERS (WIDTH=8);
ASSERT (WIDTH != 1)
REPORT "Value of parametr WIDTH = %"WIDTH
SEVERITY INFO;
ASSERT (WIDTH > 0)
REPORT "Value of parametr WIDTH must be greater than %"WIDTH
SEVERITY INFO;
SUBDESIGN DMX_P
(
  IN[WIDTH..1], ADR[3..1]      :INPUT;
  OUT[7..0][WIDTH..1]        :OUTPUT;
```

```

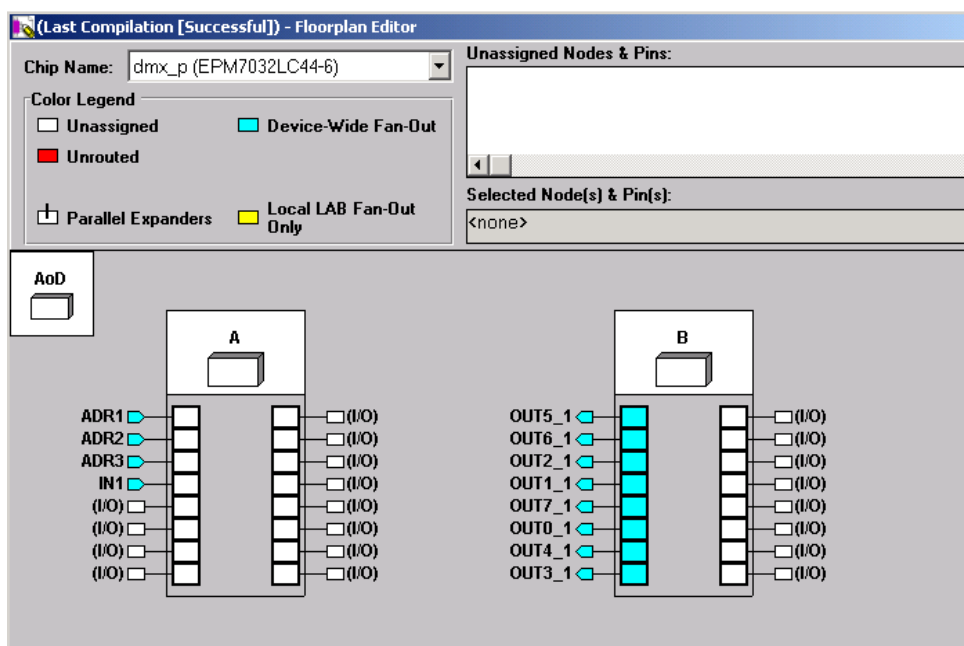
)
BEGIN
CASE ADR[] IS
WHEN 0 => OUT[0][] = IN[];
WHEN 1 => OUT[1][] = IN[];
WHEN 2 => OUT[2][] = IN[];
WHEN 3 => OUT[3][] = IN[];
WHEN 4 => OUT[4][] = IN[];
WHEN 5 => OUT[5][] = IN[];
WHEN 6 => OUT[6][] = IN[];
WHEN 7 => OUT[7][] = IN[];
END CASE;
END;

```

Parametriui WIDTH Assign lentelėje panaudojant komandą Global Project Parameters priskiriamas vienetas. Modeliavimo rezultatai atrodo taip:



5.29 pav. Parametruoto demultiplekserio modeliavimo rezultatai grafiniame režime



5.30 pav. Floorplan Editor funkcijos panaudojimas parametrizuoto demultiplekserio modeliavime

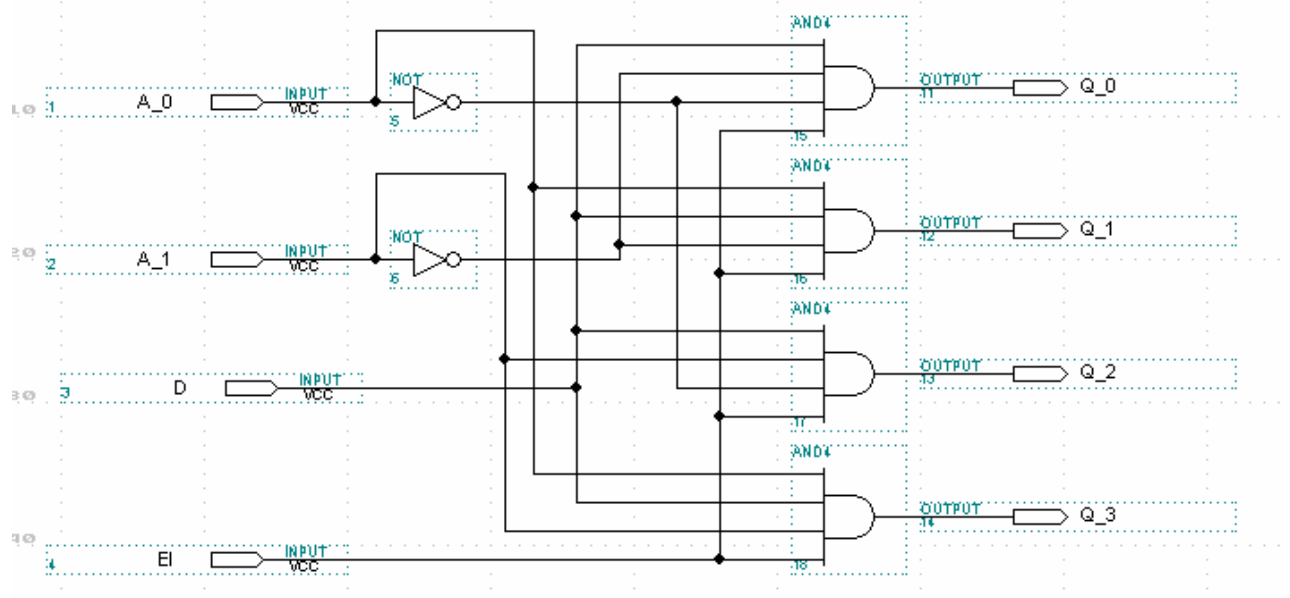
5.3.8 Demultiplekserio modeliavimas grafiniu būdu

Grafiniu būdu demultiplekserio projektavimas buvo atliktas, sukomutuojuant keturis IR elementus ir du NE elementus, nes demultiplekseriai atlieka atbulinę multiplekserio operaciją – perduoda duomenis iš vieno įėjimo kanalo į vieną iš išėjimo kanalų.

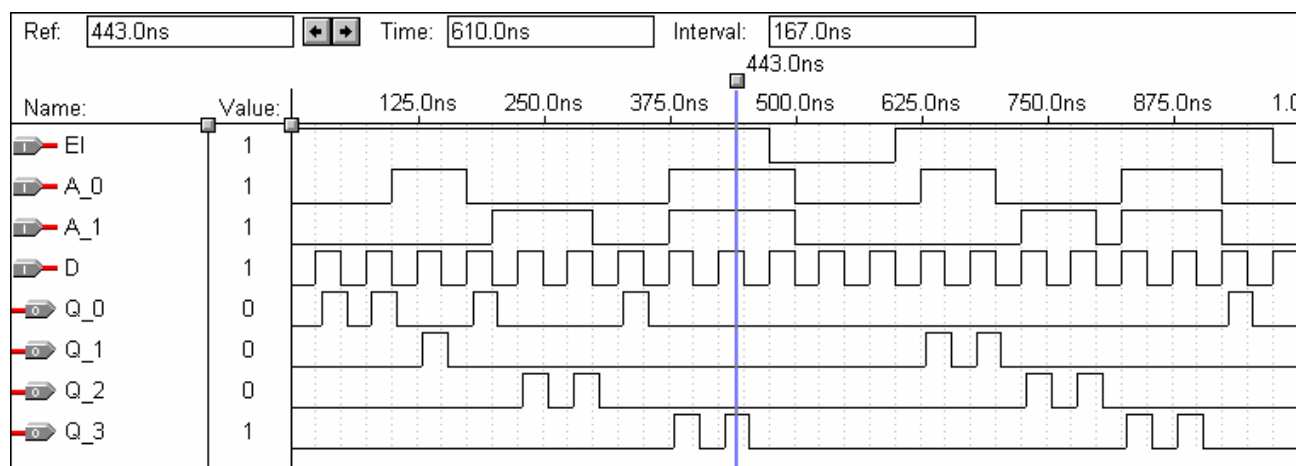
Suprojektuotas demultiplekseris turi du adresinius įėjimus, vieną darbo leidimo įėjimą ir keturis išėjimus. Jo reikšmių lentelė atrodo taip:

8.2 lentelė. Multiplekserio reikšmių lentelė

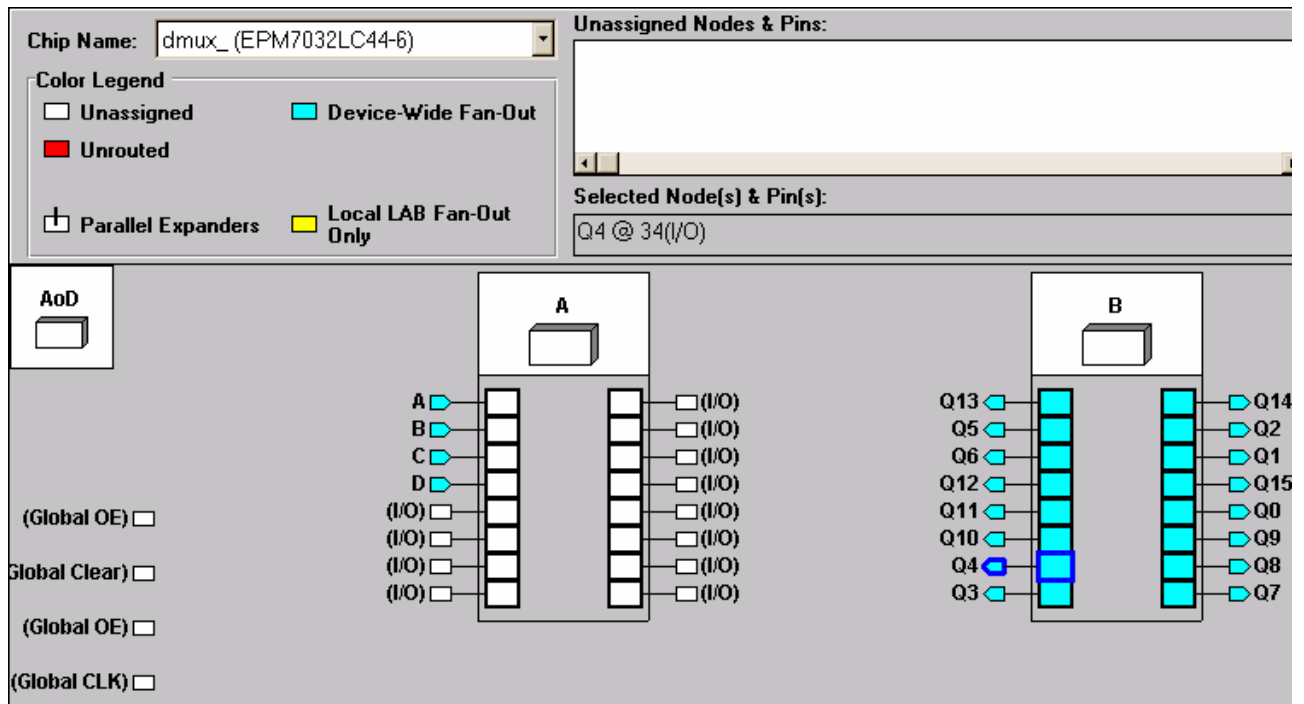
EI	A 1	A 0	Q 0	Q 1	Q 2	Q 3
0	X	X	0	0	0	0
1	0	0	D	0	0	0
1	0	1	0	D	0	0
1	1	0	0	0	D	0
1	1	1	0	0	0	D



5.31 pav. Demultiplekserio loginė schema



5.32 pav. Demultiplekserio schemos modeliavimo rezultatai



5.33 pav. Floorplan Editor funkcijos panaudojimas demultiplekserio modeliavime

5.4 Skaitiklio modeliavimas

5.4.1 Skaitiklio modeliavimas tekstiniu būdu

Skaitiklis – tai nuoseklus įtaisas, skirtas registruoti impulsų skaičių, kuris patenka į jo taktinį įėjimą.

Parametrinis dvejetainis WIDTH – skilčių asinchroninis sudėties skaitiklis turi tokius kontaktus:

- CLK – taktinių impulsų įėjimas;
- Reset – įėjimas skirtas „0“ nustatymui;
- Dout[WIDTH..1] – WIDTH skilčių skaitiklio išėjimas

Tekstinis aprašymas asinchroninio sudėties skaitiklio atrodys taip:

```
PARAMETERS (WIDTH= 8);
ASSERT (WIDTH>0)
    REPORT "Value of WIDTH parameter must be greater than %"WIDTH
    SEVERITY ERROR;
SUBDESIGN ct_sum
(
  CLK           :INPUT;
  Reset        :INPUT= GND;
  Dout[WIDTH..1] :OUTPUT;
)
VARIABLE
  RG[WIDTH..1] : DFF;
```



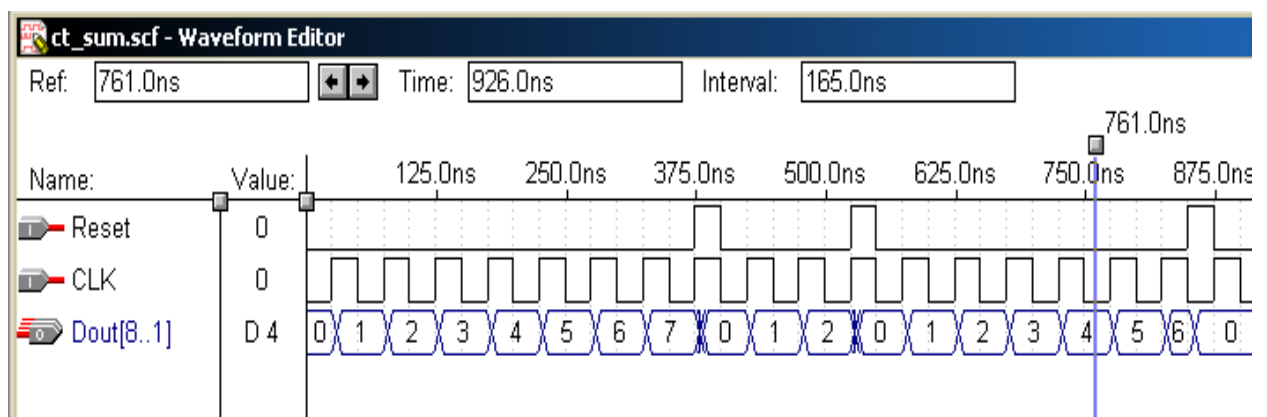
```

BEGIN
RG[].CLRn= !Reset;
  RG[].CLK= CLK;
  RG[].D= RG[].Q+1;
  Dout[]= RG[].Q;
END;

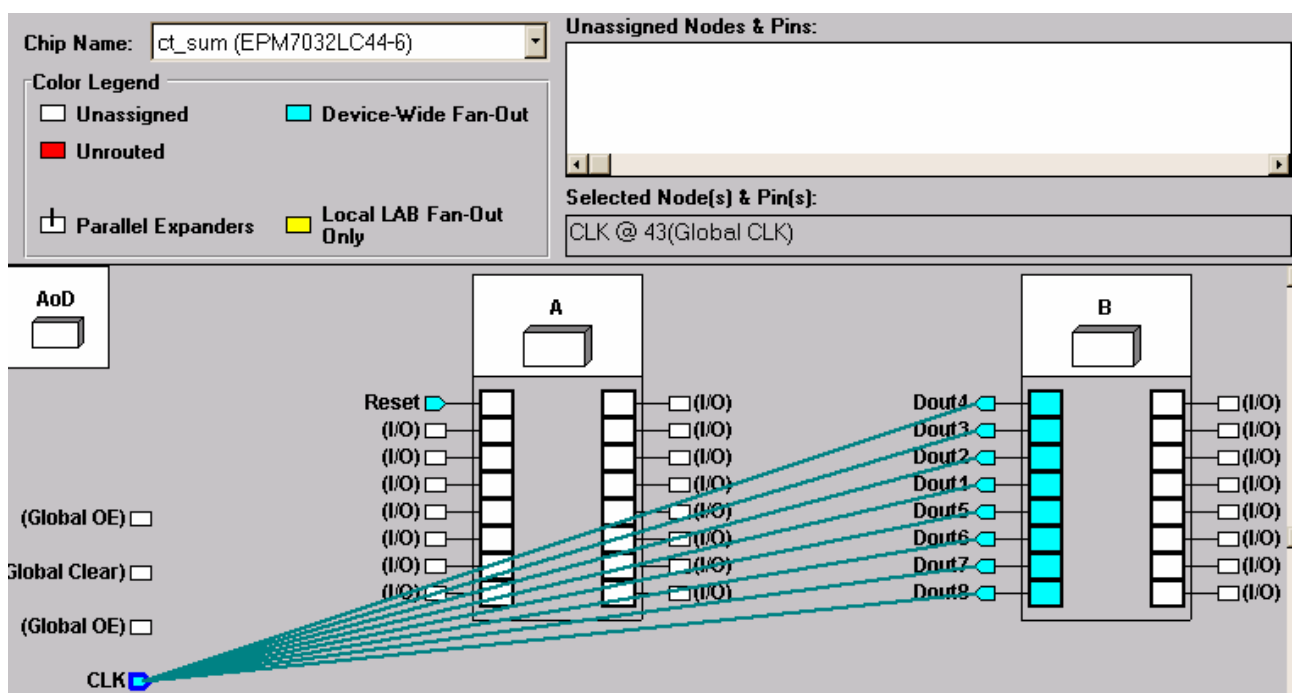
```

Tekstiniame aprašyme buvo panaudotas trigerio primityvas DFF (RG[WIDTH..1] : DFF), kurio prototipas bus FUNCTION DFF (D, CLK, CLRn, PRN) kur D – informacinis įėjimas, CLK – taktinio impulso įėjimas, CLRn – trigerio asinchroninis numetimas (aktyvus lygis yra loginis nulis), PRN – trigerio asinchroninis nustatymas (aktyvus lygis yra loginis nulis).

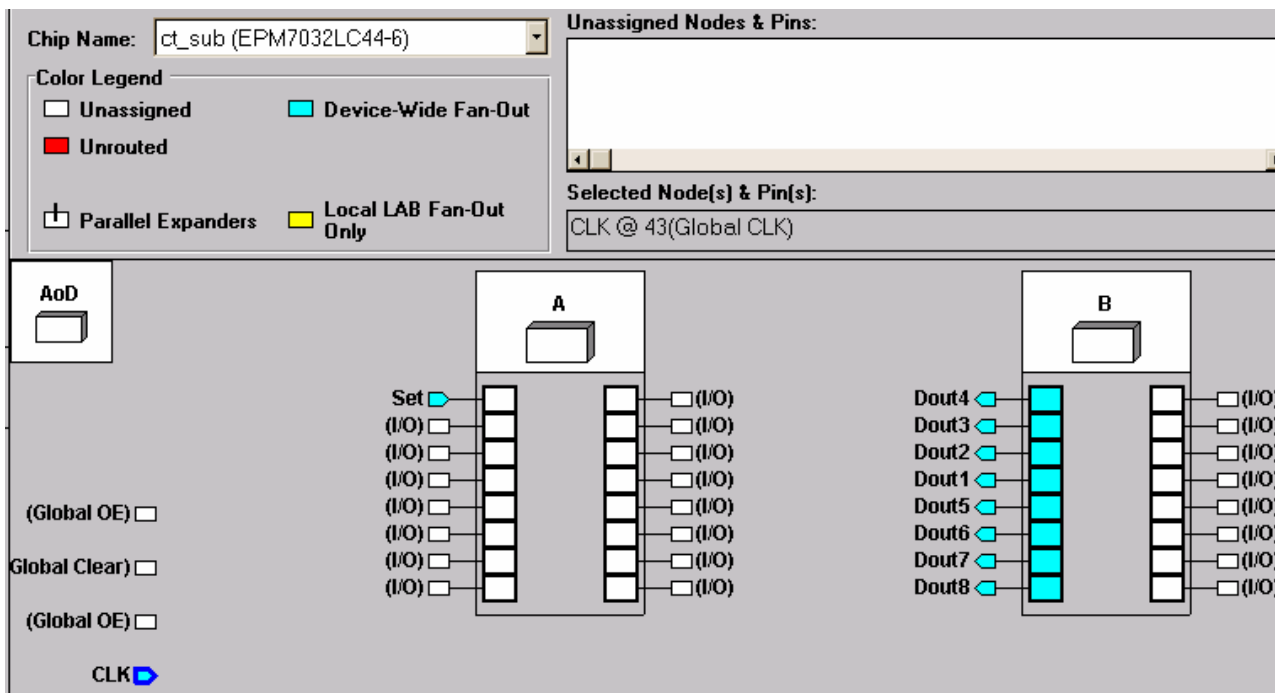
Modeliavimo rezultatai ct_sum skaitiklio kai WIDTH=8 (WIDTH parametro reikšmė buvo nustatyta Assign/Global Project Parameters) pavaizduoti 5.34 pav.



5.34 pav. Parametrinio asinchroninio sudėties skaitiklio modeliavimo rezultatai



8.35 pav. Floorplan Editor funkcijos panaudojimas parametrinio asinchroninio sudėties skaitiklio modeliavime.



5.37 pav. Floorplan Editor funkcijos panaudojimas parametrinio asinchroninio atimties skaitiklio modeliavime

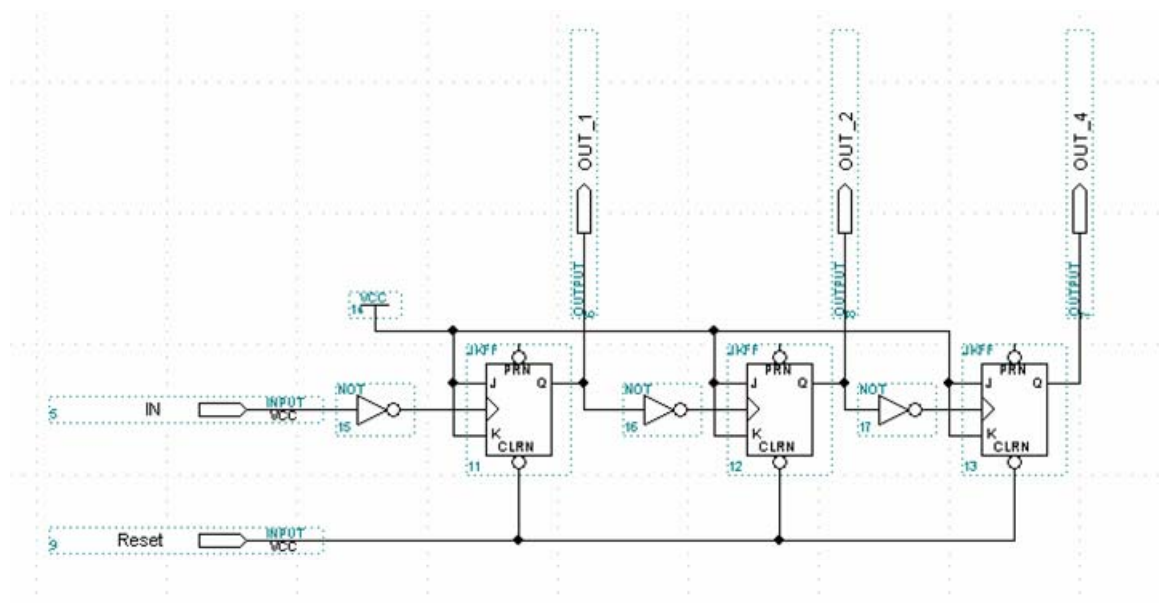
Modeliuojant asinchroninį dvejetainį WIDTH skilčių reversinį skaitiklį, kuris turi darbo leidimo įėjimą ir asinchroninio nulio nustatymo įėjimą, laikoma, kad skaitiklis turi tokius kontaktus:

- CLK – taktinių impulsų įėjimas;
- Sreset – įėjimas skirtas asinchroniniam skaitiklio nustatymui į nulio būseną;
- Dout[WIDTH..1] – WIDTH skilčių skaitiklio išėjimas
- Enable – darbo leidimo įėjimą
- Dir – įėjimas skirtas skaičiavimo valdymui

Asinchroninio reversinio skaitiklio tekstinis aprašymas atrodo taip:

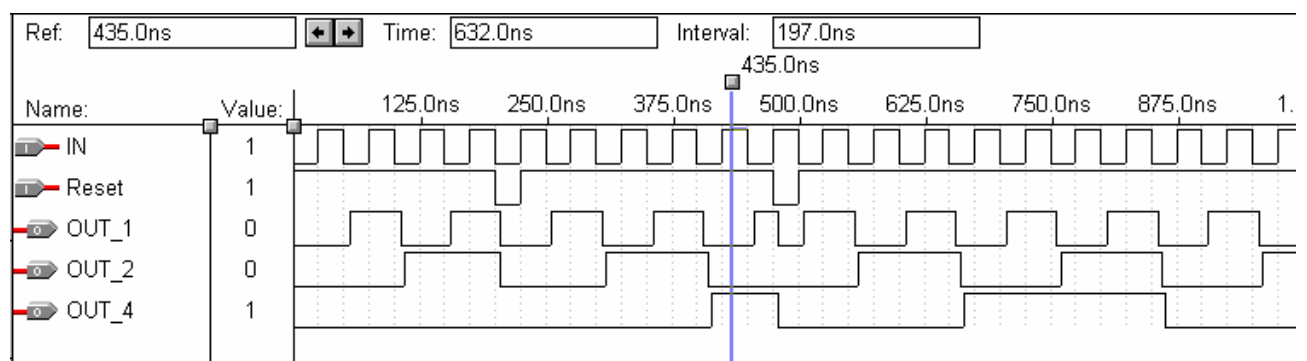
```
PARAMETERS (WIDTH= 8);
ASSERT (WIDTH>0)
    REPORT "Value of WIDTH parameter must be greater than %"WIDTH
    SEVERITY ERROR;
SUBDESIGN ct_rev
(
CLK          :INPUT;
SReset       :INPUT= GND;
Enable, Dir  :INPUT= VCC;
Dout[WIDTH..1] :OUTPUT;
)
VARIABLE
    RG[WIDTH..1] : DFFE;
BEGIN
RG[.](CLK, Ena)=(CLK, Enable);
if Sreset
```


5.4.2 Skaitiklio modeliavimas grafiniu būdu

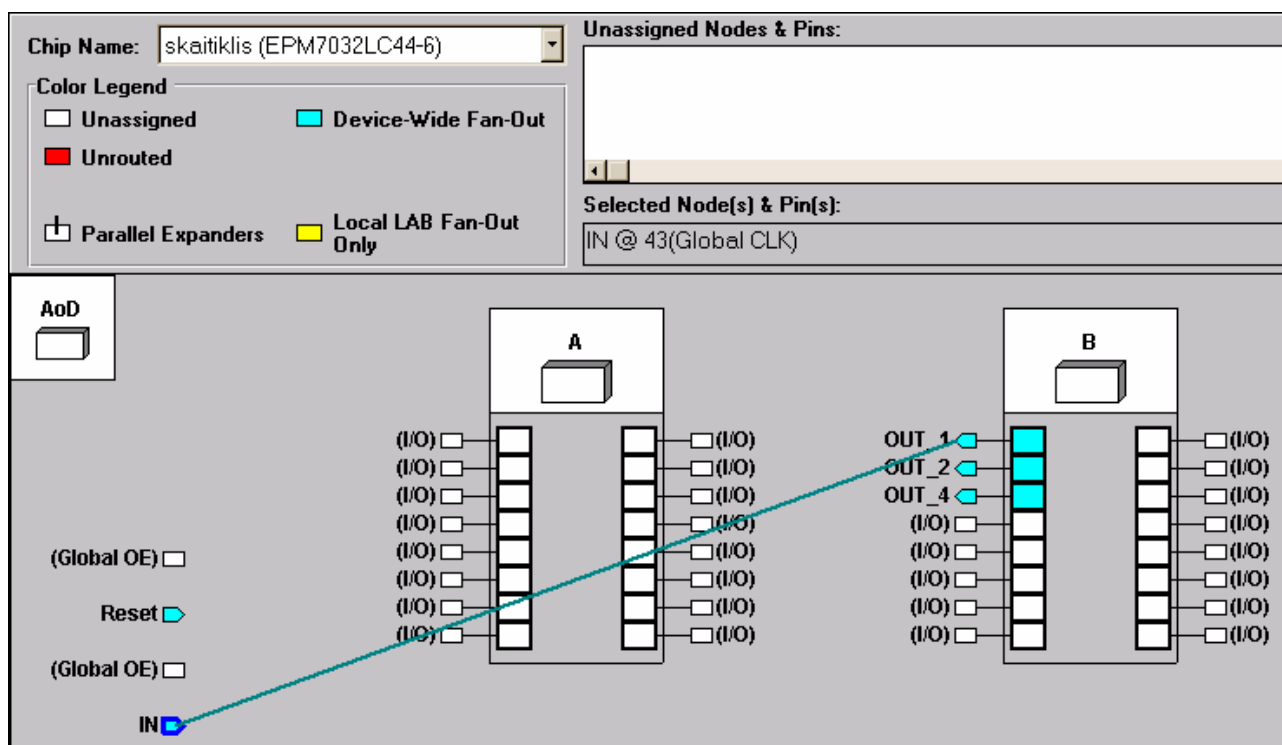


5.40 pav. Asinchroninis dvejetainis trijų skilčių sudėties skaitiklis

Asinchroninio dvejetainio sudėties skaitiklio projektavimas buvo atliekamas su JK tipo triggeriais. Loginė skaitiklio schema buvo sudaryta taip, kad: žemiausios skilties Q1 trigeris keičia būseną nuo kiekvieno įėjimo impulso, skilties Q trigeris nuo kas antro įėjimo impulso, skilties Q4 nuo kas ketvirto įėjimo impulso. Iš to seka, kad Q2 ir Q4 skilčių trigeriai keičia būseną, kai visi prieš tai esantys žemesnių skilčių trigeriai keičia būseną iš 1 į 0. Todėl projektavimo metu trigeriai turi būti valdomi su dinamiiniu inversiniu valdymu. Skaičiuojami impulsai turi būti paduoti į žemiausios skilties sinchronizacijos įėjimą.

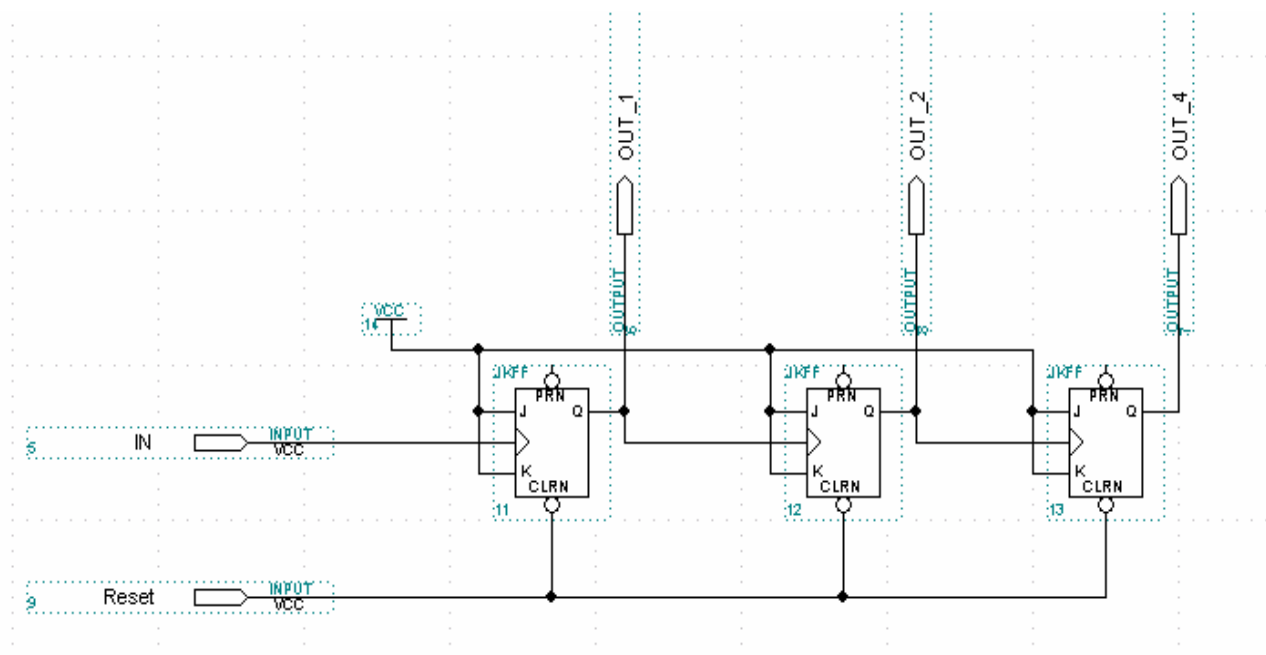


5.41 pav. Asinchroninio dvejetainio trijų skilčių sudėties skaitiklio modeliavimo rezultatai

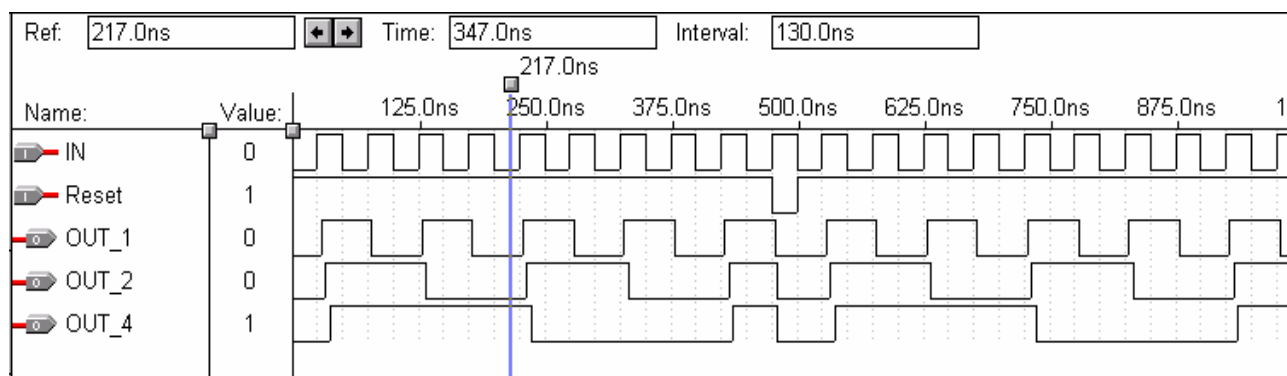


5.42 pav. Floorplan Editor funkcijos panaudojimas asinchroninio dvejetainio trijų skilčių sudėties skaitiklio modeliavime.

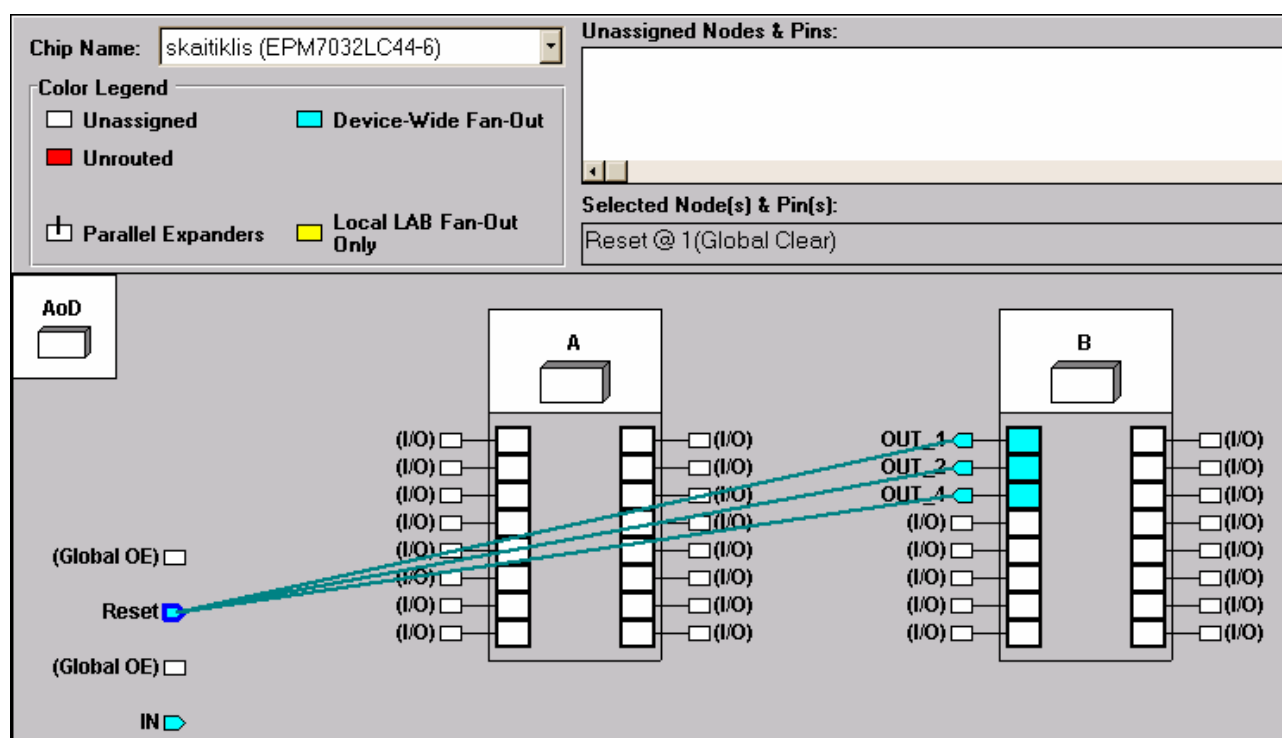
Asinchroninio dvejetainio atimties skaitiklio modeliavimas gali būti atliktas panaudojant panašios struktūros schemą. Atimties skaitiklio funkcionavimo schema skyrėsi savo pradine būsena (111 vietoje 000) ir tuo kad visų aukštesnių skilčių trigeriai pereina į kitą būseną tuomet, kai prieš jį esančios žemesnės skilties trigeris pereina iš 0 į 1 būseną, o ne atvirkščiai, kaip buvo sudėties skaitiklio atveju.



5.43 pav. Asinchroninis dvejetainis trijų skilčių atimties skaitiklis



5.44 pav. Asinchroninio dvejetainio trijų skilčių atimties skaitiklio modeliavimo rezultatai

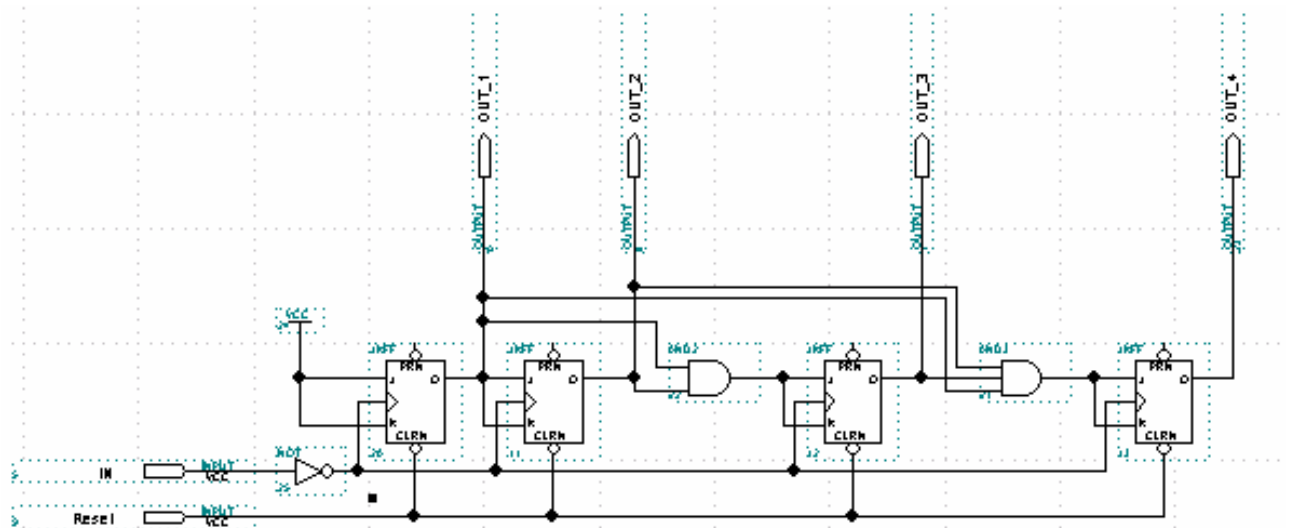


5.45 pav. Floorplan Editor funkcijos panaudojimas asinchroninio dvejetainio trijų skilčių atimties skaitiklio modeliavime

Visuose šiuose skaitikliuose aukštesnių skilčių trigerių reikšmės tampa pastoviomis tik nusistovėjus visų žemesnių skilčių trigerių reikšmėms. Jei vienos skilties trigerio vėlinimo laikas (įvertinant ir tarp trigerio esančio loginio elemento vėlinimo laiką) yra t_p , tai maksimalus n skilčių skaitiklių nusistovėjimo laikas $t_p = n t_p$ yra gana didelis.

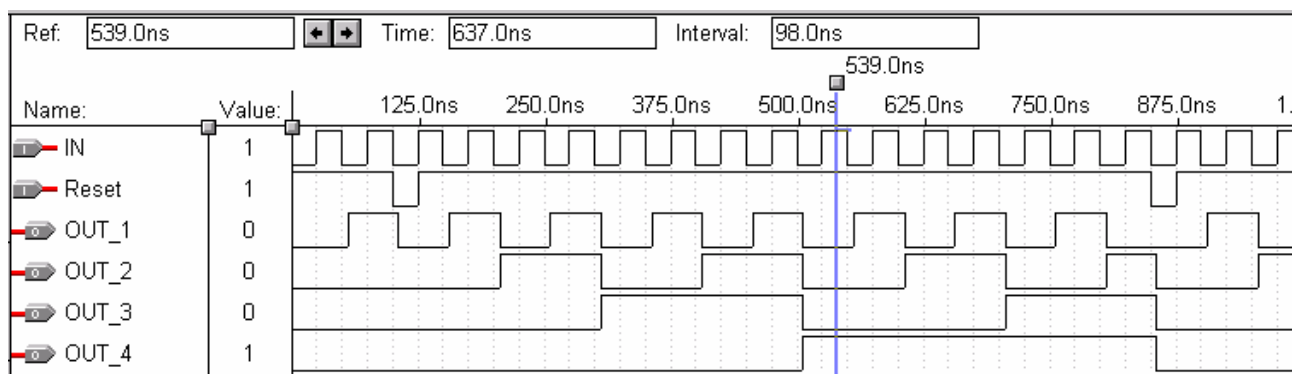
Sinchroninių lygiagreto perkėlimo skaitiklių nusistovėjimo laikas yra daug mažesnis. Juose įėjimo impulsai vienu metu, lygiagrečiai, paduodami į visų skilčių trigerius. Tokio skaitiklio vėlinimo laikas artimas vienos skilties trigerio vėlinimo laikui ir nepriklauso nuo skilčių skaičiaus.

Sinchroninio sudėties skaitiklio schemoje (5.46 pav.) panaudoti dinaminio valdymo JK trigeriai, kurių visi sinchronizacijos įėjimas sujungti ir į juos visus paduodami skaičiuojami impulsai.

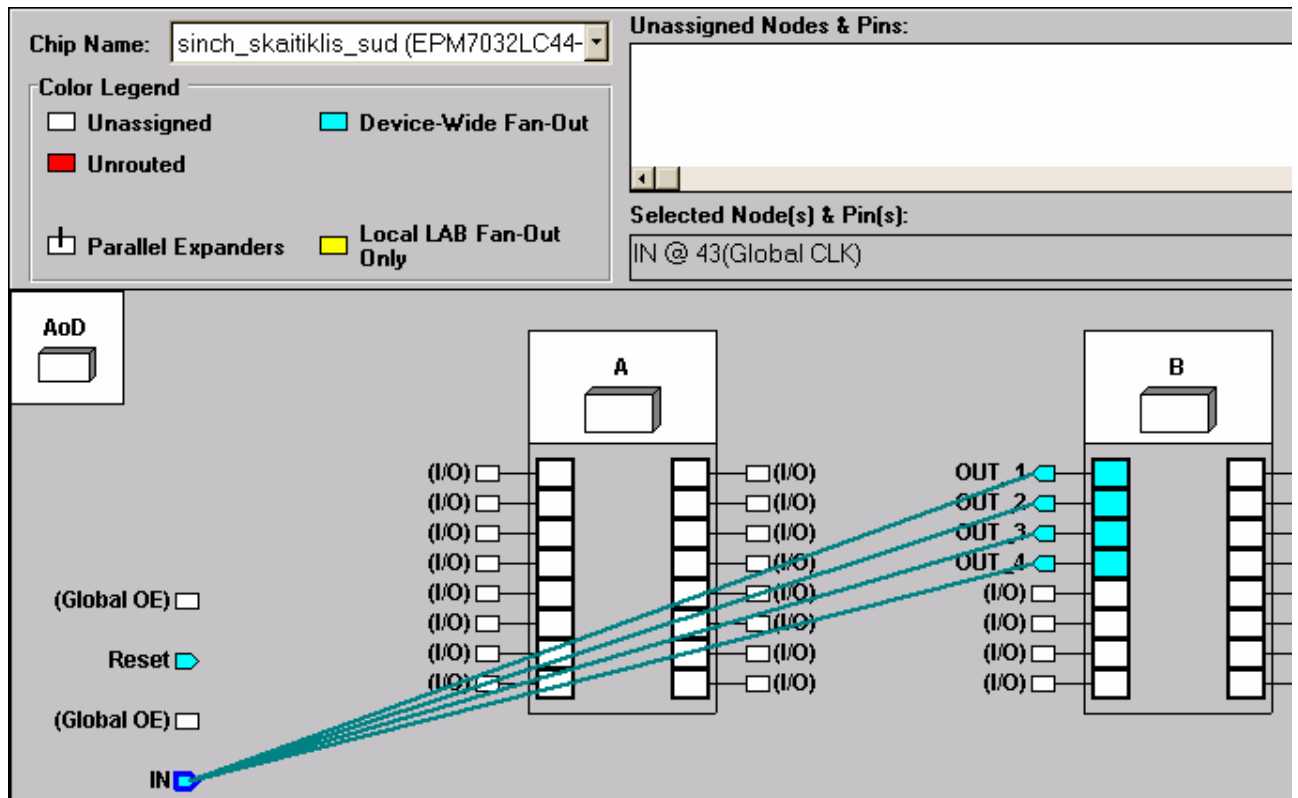


5.46 pav. Sinchroninis keturių skilčių sudėties skaitiklis

Pavaizduotame skaitiklyje žemiausios skilties trigerio JK įėjimai sujungti su maitinimo šaltiniu (VCC), todėl jis keičia būseną nuo kiekvieno skaičiuojamo impulso. Visų aukštesnių skilčių trigerių JK įėjimai sujungti prieš juos esančių trigerių tiesioginiais išėjimais per loginę schemą IR. Tai reiškia, kad skaičiavimo režime jie bus kaip tik tuomet, kai prieš juos esančių trigerių išėjimuose bus loginis 1, nes tik tuomet IR schemos išėjime bus loginis 1.



5.47 pav. Sinchroninio keturių skilčių sudėties skaitiklio modeliavimo rezultatai



5.48 pav. Floorplan Editor funkcijos panaudojimas sinchroninio keturių skilčių sudėties skaitiklio modeliavime

Daugiau informacijos ir programas galima rasti INTERNETE

6. ATERA INTERNETE

1. <http://www.ALTERA.com/literature/>
2. <http://www.ALTERA.com/products/devices/max3k/m3k-index.htm>
3. <http://www.ALTERA.ru/FAQ/FAQ.htm>
4. <http://www.ALTERA.ru/Training/Training.htm>
5. <http://www.sm.bmstu.ru/sm5/n4/oba/zan2.html>
6. <http://www.sm.bmstu.ru/sm5/n4/oba/zan4.html>
7. <http://www.sm.bmstu.ru/sm5/n4/oba/zan5.html>

7. INFORMACIJA APIE ALTERA PRODUKTUS INTERNETĖ

Altera® development kits provide a complete, high-quality design environment for engineers. A wide variety of kits help simplify the design process and reduce time-to-market. Development kits include software, reference designs, cables, and programming hardware. Table 1 lists Altera development kits.

7.1. Purchasing Information

Contact one of our [distributors](#) or [sales representatives](#) for information on how to purchase Altera products or purchase on-line from the [Altera eStore](#).

Table 1. Altera Development Kits (1)

Product Name ▲ ▼	Device ▲ ▼	Price ▲ ▼
Arria GX Development Kit	Arria GX 1AGX60DF780	\$ 995
Audio Video Development Kit, Stratix II GX Edition	Stratix II GX 2SGX90	\$4,995
Cyclone II Starter Development Kit	Cyclone 2C20	\$ 150
Cyclone III FPGA Development Kit	Cyclone 3C120N	\$1,495
Cyclone III Starter FPGA Kit	Cyclone 3C25	\$ 199
DSP Development Kit, Cyclone II Edition	Cyclone 2C70	\$ 995
DSP Development Kit, Stratix II Edition	Stratix 2S60	\$1,995
DSP Development Kit, Stratix II Professional Edition	Stratix II 2S180	\$5,995
MAX II Development Kit	MAX 1270	\$ 150
Nios II Development Kit, Cyclone II Edition (2C35)	Cyclone 2C35	\$ 995
Nios II Development Kit, Stratix II Edition	Stratix 2S60	\$ 995
Nios II Embedded Evaluation Kit, Cyclone III Edition	Cyclone 3C25	\$ 399
PCI Development Kit, Cyclone II Edition	Cyclone II 2C35	\$ 995
PCI Express Development Kit, Stratix II GX Edition	Stratix II GX 2SGX90	\$2,995
Stratix III FPGA Development Kit	Stratix III 3SL150	\$2,495
Transceiver Signal Integrity Development Kit, Stratix II GX Edition	Stratix II GX 2SGX90	\$1,295
Video Development Kit, Cyclone II Edition	Cyclone II 2C70	\$1,095

Note:

- The following product names, words, and family names are trademarks or registered trademarks of Altera Corporation: Arria™, Stratix®, Cyclone®, and MAX® devices, and Nios® processors. For a complete list of Altera trademarks and service marks go to the Altera [Legal Notice](#) web page.

7.2. Arria GX FPGA Development Kit from Altera Corporation

GX Development Kit delivers a complete environment for the development and testing of designs implementing high-speed serial interfaces in Arria GX FPGAs. This development kit is built on a PCIe form-factor card and targets the development of designs utilizing PCI Express x1 and x4, Gigabit Ethernet, and/or Serial RapidIO® (SRIO) protocols.

Ordering Information

Table 1. Arria GX FPGA Development Kit Ordering Code and Pricing Information

Ordering Code	Price	Ordering Information
DK-DEV-1AGX60N	\$995	Purchase online via Altera's eStore or contact your local Altera® distributor to place your order.

Development Kit Contents

The Arria GX FPGA Development Kit is RoHS compliant and features the following:

- Arria GX development board
 - Arria GX EP1AGX60DF780C6N FPGA
 - 60,100 equivalent logic elements (LEs)
 - 350 user I/O pins
 - 8 transceiver channels
 - PCI Express x4 edge connector
 - One high-speed mezzanine connector (HSMC) for Gigabit Ethernet [\(1\)](#) and SRIO [\(2\)](#) connectivity
 - Power and analog devices from Linear Technology
 - Dual power path controller [LTC4416EMS-1](#)
 - Low input voltage VLDO regulator [LTC3026EDD](#)
 - Low noise, fast transient response LDO regulator [LT1963AEQ](#)
 - Dual, two-phase synchronous regulator [LTC3728LEUH-1](#)
 - 32-MByte x 16 DDR2 SDRAM operating at 233 MHz
 - Switches and indicators
 - Four user-definable, push-button switches
 - Eight-position, user-definable, dual in-line package (DIP) switch
 - Eight user-definable LEDs
 - One nCONFIG push button
 - One power-on reset push button
 - Configuration circuitry
 - MAX® II/flash configuration circuit
 - JTAG download port
- Quartus® II Web Edition design software
- PCI Express Development Kit, Arria GX Edition CD ROM
 - PCI Express reference design
 - Complete documentation
 - Development Kit, Arria GX Edition *Getting Started User Guide*
 - Board reference manual
 - Board schematic and layout information
- Support for OpenCore Plus IP megafunctions
 - PCI-Express Compiler: x1 and x4
 - Triple-Speed Ethernet
 - Serial RapidIO
- Cable and accessories
 - USB-Blaster™ download cable



- External AC adapter power supply
- Power cord (including support for UK, Europe)

7.3. Cyclone II FPGA Starter Development Kit

from *Altera Corporation*

- [Ordering Information](#)
- [Development Kit Contents](#)
- [Available Documentation](#)

The low-cost Cyclone® II FPGA Starter Development Kit is ideal for evaluating Altera's high-performance, low-power, 90-nm technology. By using this RoHS compliant starter development kit, you will see 60 percent (on average) higher performance and 50 percent (on average) lower [power](#) than competing 90-nm, low-cost FPGAs. Several reference designs and demonstrations included in the kit make for a quick, "out-of-the-box" evaluation experience.

The Cyclone II FPGA Starter Development Kit includes the following:

- Altera's easy-to-use Quartus® II design software (Web Edition)
- Nios® II Embedded Design Suite (EDS)
- Complete documentation
- Cables and power supply

Cyclone II FPGAs are perfectly suited as an embedded processor or microcontroller when combined with Altera's 32-bit Nios II embedded processor intellectual property (IP) cores. You can add many other functions to the FPGA with additional IP cores available from Altera and Altera's partners.

Additional daughtercards, which plug into the Cyclone II FPGA Starter Development Kit, are available for purchase.

Ordering Information

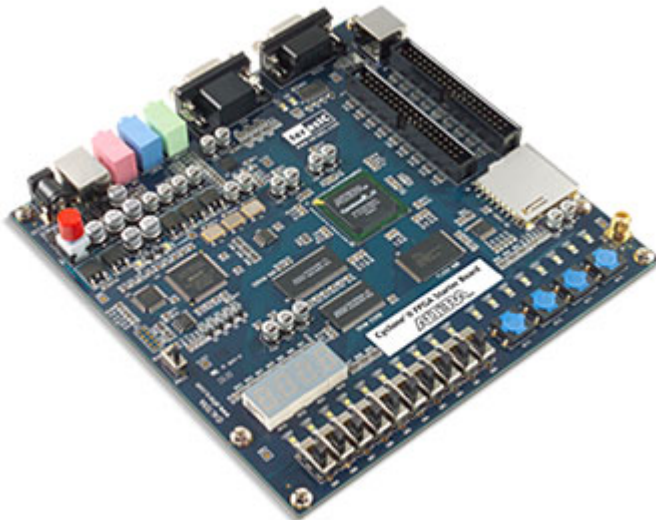
Ordering Code	Price	Ordering Information
DK-CYCII-2C20N	\$150	Purchase online via Altera's eStore or contact your local Altera® distributor to place your order.
Optional Daughtercards	\$99	3.6 Inch Digital Panel Kit: View Altera partner information or order online.
	\$70	1.3 Megapixel Digital Camera Kit: View Altera partner information or order online.

Development Kit Contents

The Cyclone II FPGA Starter Development Kit is RoHS compliant and features:

- Cyclone II Starter Development Board
- Cyclone II EP2C20F484C7N device
- Configuration
 - USB-Blaster™ download cable (embedded)
 - EPCS4 serial configuration device
- Memory
 - 8-Mbyte SDRAM
 - 512-Kbit SRAM
 - 1- to 4-Mbyte flash
- Clocking
 - SMA connector (external clock input)
- Audio
 - 24-bit coder/decoder (CODEC)
- Switches and indicators
 - Ten switches and four push buttons
 - Four 7-segment displays
 - Ten red and eight green LEDs
- Connectors
 - VGA, RS-232, and PS/2 ports
 - Two 40-pin expansion ports
 - SD/MMC socket
- Cables/power
 - USB cable
 - External power supply with U.S. adaptor (optional, but recommended when using the kit with additional accessory daughtercards)
- Cyclone II FPGA Starter Development Kit CD-ROM
 - Reference designs and demonstrations targeted for the Cyclone II FPGA Starter Development Board
 - User manual
 - Reference guide
- Quartus II Web Edition CD-ROM
- Nios II EDS CD-ROM

Figure 1. Cyclone II FPGA Starter Development Board



Available Documentation

Table 2. Document Available for the Cyclone II FPGA Starter Development Kit		
Document	File Format	Language
User Guide	Adobe PDF	English

	Adobe PDF	Japanese
	Adobe PDF	Simplified Chinese
Reference Manual	Adobe PDF	English
	Adobe PDF	Japanese
Development Board Schematic	Adobe PDF	English

Table 3. Errata for the Cyclone II FPGA Starter Development Kit

Document	File Format	Language	Comments
CII Starter NIOS.v	Adobe PDF	English	Note 1

Note:

1. *Problem*= Nios design example included with the kit will not compile or user will receive a 'compilation error' message.
Reason= File was not included on CD shipped with kit.
Resolution= Download this file and place into the following directory structure: `altera\kits\CII_Starter_Kit-v1.0.0\Examples\CII_Starter_demonstrations\CII_Starter_NIOS`

7.4. Cyclone III FPGA Starter Kit from Altera Corporation

The economical Cyclone III FPGA Starter Kit is easy to use and an ideal introduction for users who have never designed with FPGAs before. For experienced FPGA designers checking out the Cyclone III architecture, you'll love building systems leveraging the 60 percent (on average) faster performance and 50 percent lower power that Cyclone III FPGAs offer over the competition. Several design examples included in the kit make for a quick "out-of-the-box" evaluation experience.

Ordering Information

Table 1. Cyclone III FPGA Starter Kit Ordering Code and Pricing Information

Ordering Code	Price	Ordering Information
DK-START-3C25N	\$199	Buy now online via Altera's eStore or contact your local Altera® distributor to place your order.

Table 2. Optional [HSMC](#) Interface Compatible Daughtercards

Product Name	Type	Vendor	Host Board	Price/Availability
Digital Visual Interface (DVI) Input/Output	HSMC daughtercard	Bitec	Cyclone III FPGA Starter Kit	~ \$230 Buy Now ▶
Dual Channel Broadcast Video	HSMC daughtercard	Bitec	Cyclone III FPGA Development Kit	Contact Partner
Multi-Channel Video Input	HSMC daughtercard	Bitec	Cyclone III FPGA Development Kit	Contact Partner
HSMC to Dual Santa Cruz Adapter	HSMC daughtercard	SLS	Any HSMC-based host board	\$89 Buy Now ▶
HSMC to Santa Cruz Adapter	HSMC daughtercard	Terasic	Any HSMC-based host board	\$70 Buy Now ▶

HSMC to GPIO Adapter	HSMC daughtercard	Terasic	Any HSMC-based host board	\$50 Buy Now ▶
HSMC Breakout Adapter	HSMC daughtercard	Bitec	Any HSMC-based host board	~ \$70

Cyclone III FPGAs are the first low-cost FPGA family available in the marketplace harnessing the cost and low-power advantages of 65-nm process technology.

Cyclone III and Nios II 32-bit soft IP processor

Cyclone III FPGAs are perfectly suited as embedded processors or microcontrollers when combined with Altera's 32-bit [Nios II](#) embedded processor intellectual property (IP) cores. You can add many other functions to the Cyclone III FPGA with additional IP cores available from Altera and Altera's partners.

High-Speed Mezzanine Card Interface

Altera developed the specification for the high-speed mezzanine connector (HSMC), which is based on the [Samtec mechanical connector](#), to define and standardize the interface between optional daughtercards and host boards. This specification outlines both the electrical and mechanical properties of the interface between daughtercard and host. You can also create your own HSMC interface compatible daughtercards.

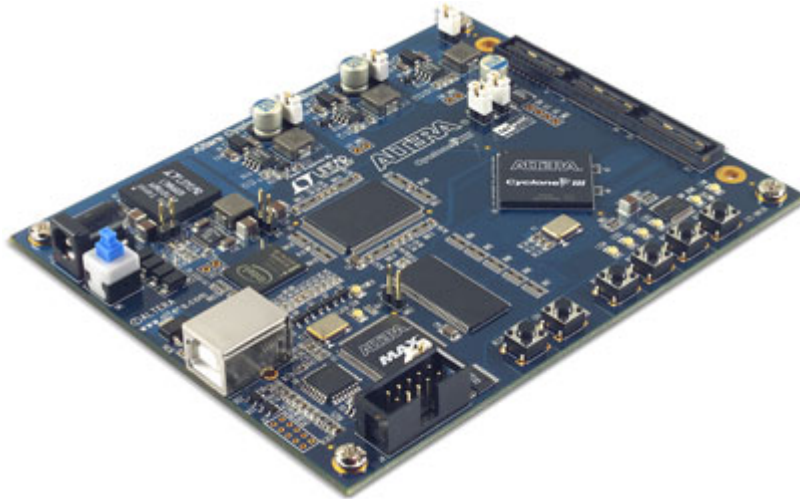
Development Kit Contents

The Cyclone III FPGA Starter Development Kit is RoHS compliant and features:

- Cyclone III starter board (see Figure 1)
 - Cyclone III EP3C25F324 FPGA
 - Configuration
 - Embedded USB-Blaster™ circuitry (includes an Altera EPM3128A CPLD) allowing download of FPGA configuration files via the user's USB port
 - Power and analog devices from Linear Technology
 - Switching power supply LTM4603EV-1
 - Switching and step-down regulators LTC3413 and LT1959
 - Memory
 - 256-Mbit of DDR SDRAM
 - 1-Mbyte of synchronous SRAM
 - 16-Mbytes of Intel P30/P33 flash
 - Clocking
 - 50-MHz on-board oscillator
 - Switches and indicators
 - Six push buttons total, four user controlled
 - Seven LEDs total, four user controlled
 - Connectors
 - HSMC
 - USB Type B
 - Cables and power
 - USB cable
 - External power supply (U.S. compatible plug with EU and UK adaptors)
- Cyclone III FPGA Starter Kit CD-ROM
 - Example designs targeting the Cyclone III FPGA starter board
 - Create an FPGA design in one hour
 - Power measurements of a Cyclone III FPGA
 - A 32-bit soft processor system inside an FPGA
 - Complete documentation
 - User guide
 - Reference manual
 - Board schematic and layout
 - Bill of materials
 - Product and partner information

- Download instructions to receive the latest version of the following software (at no charge):
 - Quartus® II Web Edition (FPGA design software)
 - ModelSim®-Altera Web Edition (FPGA simulation software from ModelSim)
 - Nios II Embedded Design Suite, Evaluation Edition (32-bit microprocessor software)

Figure 1. Cyclone III FPGA Starter Board



Available Documentation

Table 3. Documents Available for the Cyclone III FPGA Starter Kit

Document	File Format	Download	Language
User Guide	Adobe PDF	Via FTP	English
Reference Manual	Adobe PDF		
Board Assembly	Adobe PDF		
Board Mechanicals	Adobe PDF		
Board Schematic	Adobe PDF		
Bill of Materials	Microsoft Excel		

7.5. DSP Development Kit, Cyclone II Edition

from Altera Corporation

The DSP Development Kit, Cyclone® II Edition is RoHS compliant and delivers a complete development environment for digital signal processing (DSP). The DSP Development Kit, Cyclone II Edition provides a low-cost hardware platform for developing high-performance DSP designs based on Altera® Cyclone II FPGAs. The kit includes the Cyclone II DSP development board, DSP Builder development tool, Quartus® II development software, The MathWorks' MATLAB/Simulink evaluation software, evaluation intellectual property (IP) cores, and system reference designs and labs, as well as power supplies, cables, and documentation.

- [Ordering Information](#)
- [Development Kit Contents](#)
- [Available Documentation](#)

Ordering Information

Table 1. DSP Development Kit, Cyclone II Edition Ordering Code and Pricing Information

Ordering Code	Price	Ordering Information
DK-DSP-2C70N	\$995	In North America, call 1-888-800-0631. For international sales, contact your local distributor .
DC-VIDEO-TVP5146	\$195	In North America, call 1-888-800-0631. For international sales, contact your local distributor .

Development Kit Contents

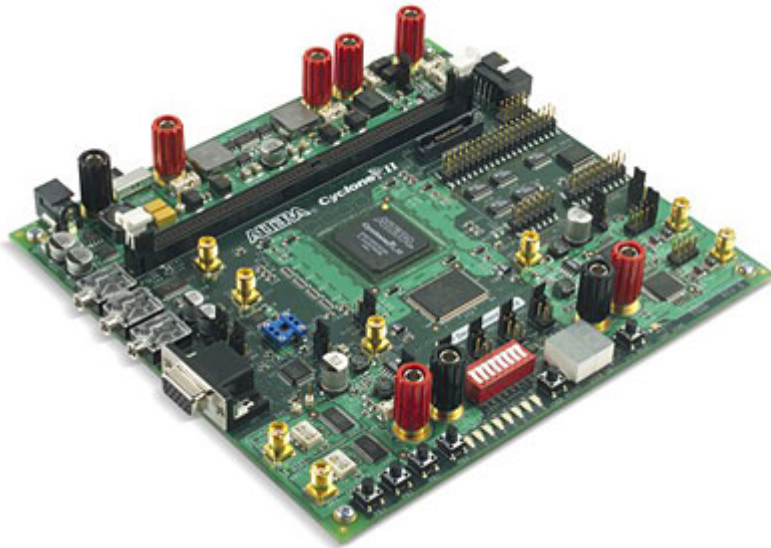
The DSP Development Kit, Cyclone II Edition features:

- Cyclone II EP2C70 DSP development board
 - Cyclone II EP2C70F672C6 device
- Analog I/O
 - Two 14-bit analog-to-digital (A/D) converter channels with 125-million samples per second (MSPS)
 - Two 14-bit, 165-MSPS, 70 dB digital-to-analog (D/A) converters
 - One 24-bit RGB VGA adapter with a DB-15 connector
 - One Audio CODEC with input, output, and amplified output
- External I/O
 - Mictor connector for hardware and software debugging
 - 3.3-V/5-V tolerant Altera expansion/prototype headers
 - One Texas Instruments Evaluation Module (TI-EVM) expansion connector to connect to the Spectrum Digital 'DSP Starter Kit (DSK) for the TMS320C6416', Revision E
- Memory subsystem
 - 256-Mbyte DDR2 SDRAM DIMM
 - 1-Mbyte synchronous SRAM (SSRAM)
 - Two EPCS64 devices
- MATLAB/Simulink evaluation software
- DSP Builder development tool
- Quartus II Education Kit
- Evaluation IP cores
- System reference designs and labs
 - DSP Builder filtering design
 - DSP Builder/SOPC Builder image processing reference design
 - Fast Fourier transform (FFT) coprocessor reference design for Texas Instruments' TMS320C6416 DSK
 - Nios® II reference designs
 - DDR2 DIMM reference design
- Cables and accessories
- USB-Blaster™ download cable
- Power supply
- International power cords

Video input daughtercard features:

- Two composite video input channels using the TI TVP5146 ADC
- Support for NTSC/PAL
- 10-bit BT.656 output
- Compatibility with expansion connector, standard on most Altera development kits and included with the Video Development Kit, Cyclone II Edition

Figure 1. Cyclone II EP2C70 DSP Development Board



Available Documentation

Document	Comments
Cyclone II EP2C70 DSP Development Board Reference Manual	PDF in English
DSP Development Kit, Cyclone II Edition Getting Started User Guide	PDF in English
Cyclone II EP2C70 DSP Development Board Schematic	PDF in English
Cyclone II EP2C70 DSP Development Board Design Files	

MAX II Development Kit from Altera Corporation

Altera's MAX[®] II Development Kit comes with a complete design environment. The kit enables users to evaluate the MAX II feature set or begin prototyping a design prior to receiving custom hardware. It includes all software, cables, and accessories needed to ensure an easy and productive evaluation of the MAX II CPLD.

Ordering Information

Ordering Code	Price	Ordering Information
DK-MAXII-1270N	\$150	ROHS-Compliant - Available in Europe only.
MAXII-DEVKIT-1270	\$150	Order online or, in North America, call 1-888-800-0631; for international sales, contact your local distributor . Not RoHS-complaint – Not available in Europe

Development Kit Contents

The MAX II Development Kit includes the following:

- MAX II Development Board
 - MAX II EPM1270F256C5 CPLD
 - USB media access control (MAC) with physical layer (PHY) and Type B connector
 - PCI Edge connector (3.3- and 5-V tolerant)

- LCD module
- SRAM (128K x 8 bit)
- Temperature gauge with serial peripheral interface (SPI)
- Onboard power meter
- Active I/O sense circuitry
- One 3.3-V-tolerant expansion/prototype header (41 available user I/O pins)
- JTAG connectors
- Four user-defined, push-button switches
- Four user-defined LEDs
- Quartus® II Web Edition design software
- Cables and accessories
 - ByteBlaster™ II parallel download cable
 - Type A-B USB cable (3 feet)
- Reference designs and demos for MAX II including:
 - USB reference design
 - PCI reference design
 - Low power demo
 - Real-time in-system programmability (ISP) demo



7.6. Nios II Development Kit, Cyclone II Edition

from Altera Corporation

Altera's Nios®II Development Kit, Cyclone® II Edition provides everything needed for system-on-a-programmable-chip (SOC) development. Based on Altera's [Nios II family of embedded processors](#) and the low-cost [Cyclone II](#) EP2C35

device, this development kit provides an ideal environment for developing and prototyping a wide range of price-sensitive



embedded applications.

Ordering Information.

You can order the Nios II Development Kit, Cyclone II Edition today (see Table 1).

Ordering Code	Price	Ordering Information
DK-NIOS-2C35N	\$995	In the United States and Canada, call 1-888-800-0631. Outside the United States and Canada, contact your local distributor .

Development Kit Contents

The Nios II Development Kit, Cyclone II Edition includes the following:

- Nios II family of embedded processors
- Nios II development tools, including
 - Nios II integrated development environment (IDE) and debugger
 - GNU tools
 - Nios II instruction set simulator (ISS)
- MicroC/OS-II real-time operating system evaluation
- [Nios II C-to-Hardware Acceleration Compiler evaluation](#)
- [NicheStack TCP/IP Network Stack - Nios II Edition evaluation](#)
- Quartus®II design software, including the SOPC Builder system development tool
 - One-year license
 - Windows platform only
- Library of standard microprocessor peripherals
- RoHS-compliant development board (shown in Figure 1), featuring [MorethanIP](#)
 - Cyclone II EP2C35F672 device
 - MAX®EPM7128AE CPLD configuration control logic
 - 2-Mbyte synchronous SRAM
 - 16-Mbyte DDR SDRAM
 - 16-Mbyte flash memory
 - EPCS64 serial configuration device (64 Mbit)
 - Eight user-defined LEDs
 - Dual seven-segment LED display
 - Power-on reset circuitry
- 10/100/1000 PHY Daughter Board with Marvell 88E1111 Ethernet PHY Device from [MorethanIP](#)
- OpenCore Plus evaluation of the [Altera® Triple-speed Ethernet IP Core](#) and supporting [design example](#)
- Cables and accessories
 - USB-Blaster™ download cable
 - One serial cable (RS-232)
 - 9-V power supply

- International power cords
- LCD module
- Ethernet (RJ-45) cable (7 ft.)
- Ethernet crossover adapter
- Many hardware and software reference designs targeted to the Nios development board, Cyclone II edition
- One-year subscription to Nios II processor-related tool upgrades. (This subscription does not include Quartus II software updates. The [Altera Software Subscription](#) includes the Quartus II design software.)
- There are no license or royalty fees connected with the Nios II development kit used to develop with Altera FPGAs and HardCopy® series structured ASICs
- Coupon for a 20 percent discount on any one instructor-led [Altera Technical Training course scheduled in North America](#)

Software Development Tools

Software engineers do not need any prior FPGA design experience to use the Nios II Development Kit, Cyclone II Edition. Altera's [Nios II Embedded Design Suite](#) provides a complete embedded software design environment for the Nios II family of embedded processors, with facilities for:

- C/C++ code entry and editing
- Project management and compilation
- Software debugging
- Flash programming

World-class development tools and complete documentation as well as multiple reference designs and a step-by-step software development tutorial (available in the online help browser) get you up and running immediately. You can write, compile, and run software on your Nios development board within minutes of opening the box.

The Nios II ISS allows you to begin developing programs before the target hardware platform is ready. The Nios II IDE lets you run programs on the ISS as easily as running them on a real hardware target.

Altera provides ports of the MicroC/OS-II real-time operating system (RTOS) and the NicheStack TCP/IP Network Stack, Nios II Edition. The MicroC/OS-II RTOS is built on the thread-safe Hardware Abstraction Layer (HAL) system library and implements a simple, well-documented RTOS scheduler. The TCP/IP stack is built on MicroC/OS-II and implements the standard UNIX Sockets application program interface (API).

Documented software example designs are provided to demonstrate all prominent features of the Nios II processors and the development environment.

Hardware Development Tools.

This Nios II development kit is a complete design environment for hardware engineers. You can create your own custom embedded system within hours by using the world-class FPGA design tools, example projects, complete documentation, and step-by-step tutorial included in the kit.

The Quartus II design software allows you to process multi-million gate designs, streamline development flows, and decrease time-to-market. The Nios II Development Kit, Cyclone II Edition includes the Quartus II design software with a one-year non-perpetual license.

Although a one-year Quartus II license is included with the kit, we recommend that you purchase a full [Altera design software subscription](#) in order to receive Quartus II software upgrades.

[SOPC Builder](#), integrated within the Quartus II software, is an automated system development tool that dramatically simplifies the task of creating high-performance SOPC designs.

Document	Comment
Nios Development Board Reference Manual, Cyclone II Edition	PDF in English
Nios II Development Kit Getting Started User Guide	PDF in English
Nios Development Board Cyclone II 2C35 Schematic	Zip File
Nios Development Board Cyclone II PCB Layout Files	Zip File
Nios Development Board Cyclone II PCB Layout Bill of Materials	Excel File

7.7. PCI Development Kit, Cyclone II Edition

from Altera Corporation

Altera's PCI Development Kit, Cyclone® II Edition provides a flexible, low-cost FPGA development platform in a PCI short card form factor. With a universal (3.3-V and 5-V) PCI edge connector, the PCI Development Kit, Cyclone II Edition plugs into any standard PCI or PCI-X slot. The kit includes both a hardware reference design and Windows software drivers to accelerate development. Users can perform transactions between the PCI interface and DDR2 memory within minutes of opening the box. The kit can be used as a development platform for 32-bit and 64-bit PCI, PCI-X, DDR2 SDRAM, and Altera Megafunction Partners Program (AMPPSM) partner intellectual property (IP) cores.

- [Ordering Information](#)
- [Development Kit Contents](#)

Ordering Information

Table 1. PCI Development Kit, Cyclone II Edition Ordering Code&Pricing Information		
Ordering Code	Price	Ordering Information
DK-PCI-2C35N	\$995	In North America, call 1-888-800-0631; for international sales, contact your local distributor .

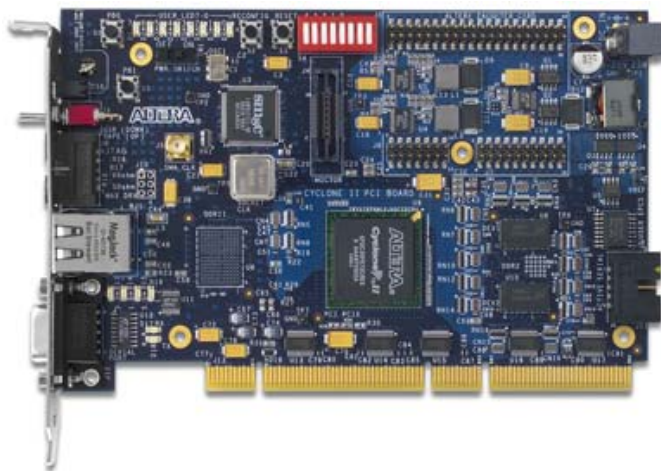
Development Kit Contents

The PCI Development Kit, Cyclone II Edition features:

- Cyclone II PCI Development Board (see Figure 1)
 - Cyclone II EP2C35F672 FPGA
 - Short-form universal PCI (3.3 or 5.0 V) card
 - 32-bit or 64-bit PCI per the PCI Local Bus Specification Revision 2.3 at 33 or 66 MHz
 - 100 MHz PCI-X Revision 2.0 mode 1
 - Memory
 - Two 64 Mbyte DDR2 SDRAM devices
 - FPGA device configuration
 - Switch-selectable on power-up, choose one of two serial configuration devices (EPCS64 devices). One device contains the pre-loaded factory default design, and the other device is for user-programming. Configuration data is downloaded via the USB-Blaster™ download cable.
 - Flexible clocking options
 - Socketed 100-MHz high-speed clock oscillator
 - SMA connector clock input
 - PCI edge connector clock input
 - Switches and indicators
 - Four user-definable push-button switches
 - Eight-position user-definable dual in-line package (DIP) switch
 - Eight user-definable LEDs
 - Flexible power options
 - PCI connector

- External power supply
- PCI Development Kit, Cyclone II Edition CD-ROM
 - PCI-to-DDR2 Reference Design
 - Cyclone II PCI Development Kit Application & Drivers
- PLD Applications PCI-X CORE CD-ROM
 - PLD Applications PCI-X function for OpenCore® evaluation
 - Reference designs&applications targeted to the Cyclone II PCI development board
- Quartus® II design software, including the SOPC Builder system development tool
 - One-year license
 - Windows platform only
- Jungo WinDriver Development Toolkit
- Cables&accessories
 - USB Blaster download cable
 - Power cable
- Complete documentation

Figure 1. Cyclone II PCI Development Board



Software Development Tools

The [Quartus II design software](#) allows designers to process multi-million-gate designs, streamline development flows, and decrease time-to-market. The PCI Development Kit, Cyclone II Edition includes the Quartus II design software with a one-year license. Altera recommends that designers purchase a full [Altera® design software subscription](#).

The kit also includes the MegaCore® IP Library CD containing all released Altera intellectual property (IP) MegaCore functions, optimized for Altera devices, including:

- PCI Compiler 32-Bit PCI Master/Target
- PCI Compiler 32-Bit PCI Target
- PCI Compiler 64-Bit Master/Target
- PCI Compiler 64-Bit Target
- DDR&DDR2 SDRAM Memory Controller Compiler

Hardware Development Tools

The PCI Development Kit, Cyclone II Edition includes VHDL and Verilog reference designs, example projects, and complete documentation.