

Įvadas į programavimą

Darbo tikslas

Susipažinti su Quartus programine įranga ir išmokti modeliuoti loginių elementų veikimą panaudojant „waveform simulation“ įrankį.

Darbo tvarka

- Užduotys turi būti atliekamos nuosekliai. Dėstytojų rodoma bei ginama tik pilnai atlikta užduotis.
- Kiekvienas laboratorinis darbas (LD) bus vertinamas iki 10 balų. LD sudaro darbo užduotys, kurias įvykūžius laiku (pagal kalendorinį planą) yra vertinama iki **10 balų**.
- Atsilikus nuo kalendorinio plano yra atliekami kiti užduočių variantai.
- Užduotys, įvykdytos atsilikus nuo kalendorinio plano, vertinamos tik iki **7 balų**.
- Darbo ataskaitos nereikia rengti, jei išsaugoma elektroninė užduoties sprendinio versija.

Darbo užduotys:

1. Paleiskite Quartus 13.1 programą. Sukursime naują projektą.

New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

C:\altera\13.1

What is the name of this project?

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

Use Existing Project Settings...

< Back Next > Finish Cancel Help

2. Sukurkite ir nurodykite jūsų asmeninį katalogą, kur bus saugomi projektai. Pasirinkite projekto pavadinimą, ir **entity** pavadinimą. **Entity** pavadinimas turės būti nurodytas jūsų programos kode.

New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

C:/Users/50562/Documents/KA_labs

What is the name of this project?

lab1


What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

lab1

Use Existing Project Settings...

< Back Next > Finish Cancel Help

3. Spaudžiame „next“ nekeisdami pradinių nustatymų. Pasitikrinkite ar nurodytas simuliacijos įrankis „ModelSim-Altera“, bei nustatyta VHDL kalba.

 New Project Wizard

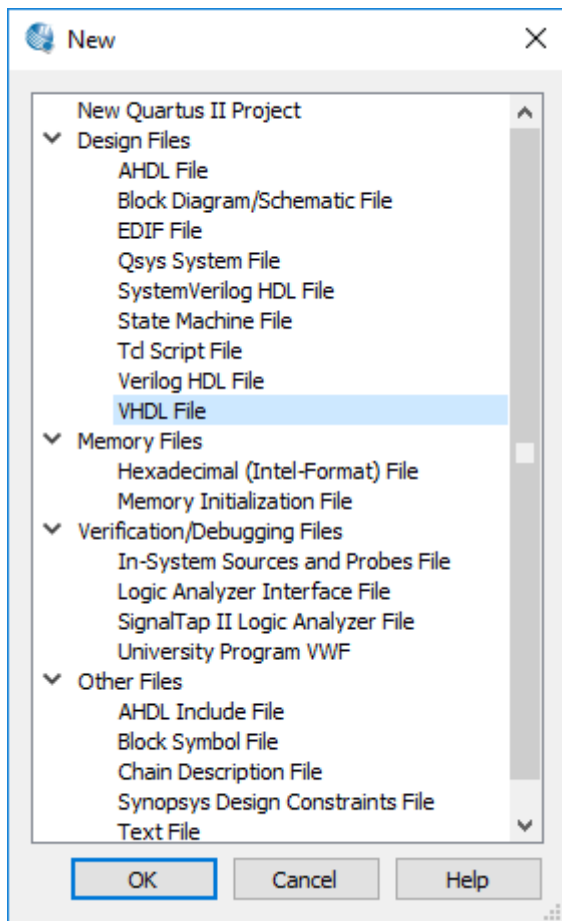
EDA Tool Settings [page 4 of 5]

Specify the other EDA tools used with the Quartus II software to develop your project.

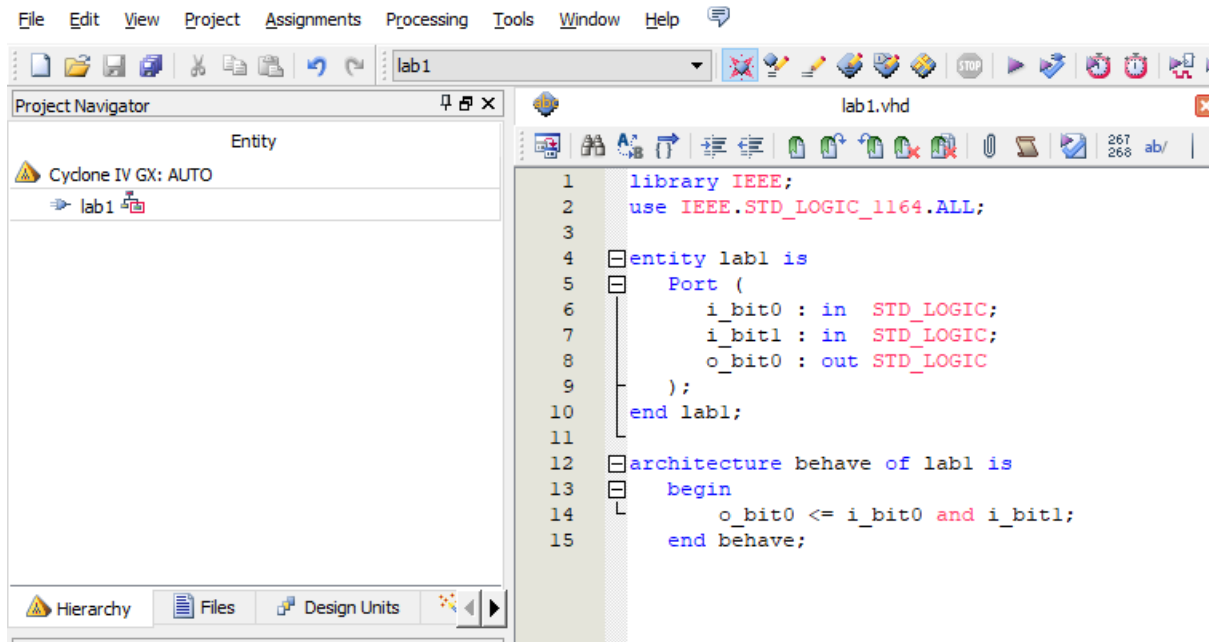
EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	VHDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

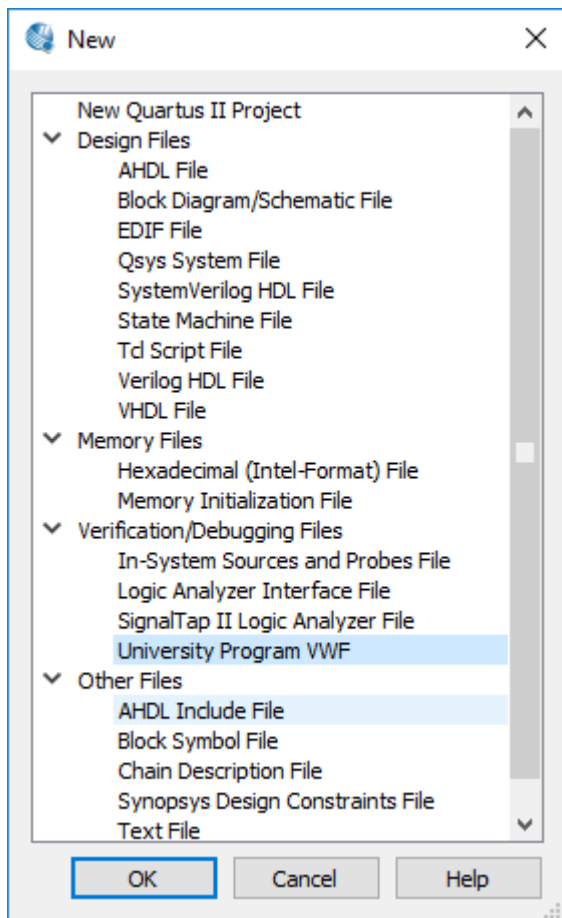
4. Jūsų projekto katalogas yra tuščias, sukurkite naują VHDL failą.



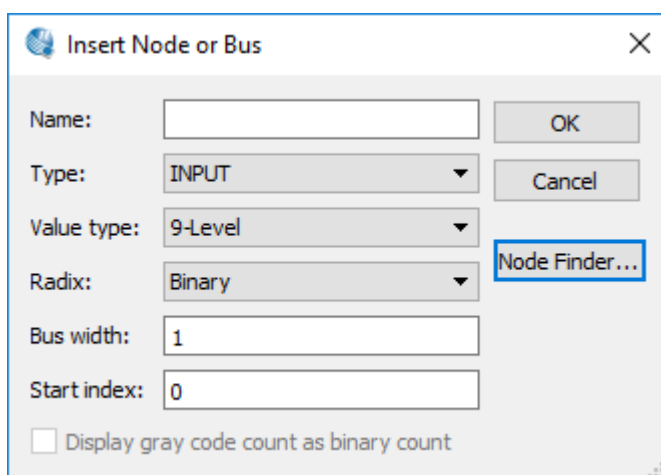
5. Aprašome naudojamas bibliotekas, sukuriame entity, kuriame bus aprašomas grandinės funkcionalumas įėjimų ir išėjimų lygmenyje. Svarbu, kad jūsų entity pavadinimas atitiktų tą, kurį nurodėte kurdami projektą. Atkreipkite dėmesį, kad paskutinis išėjimas neužsibaigia kabliataškiu. Toliau aprašoma kuriamos grandinės architektūra, ją galima pavadinti, pvz „behave“. Architektūros aprašyme yra 1 eilutė, kuri aprašo, kaip sąveikaus įėjimai ir išėjimai, koks signalas bus siunčiamas į išėjimą. Pavyzdyje aprašytas IR loginio elemento veikimas.



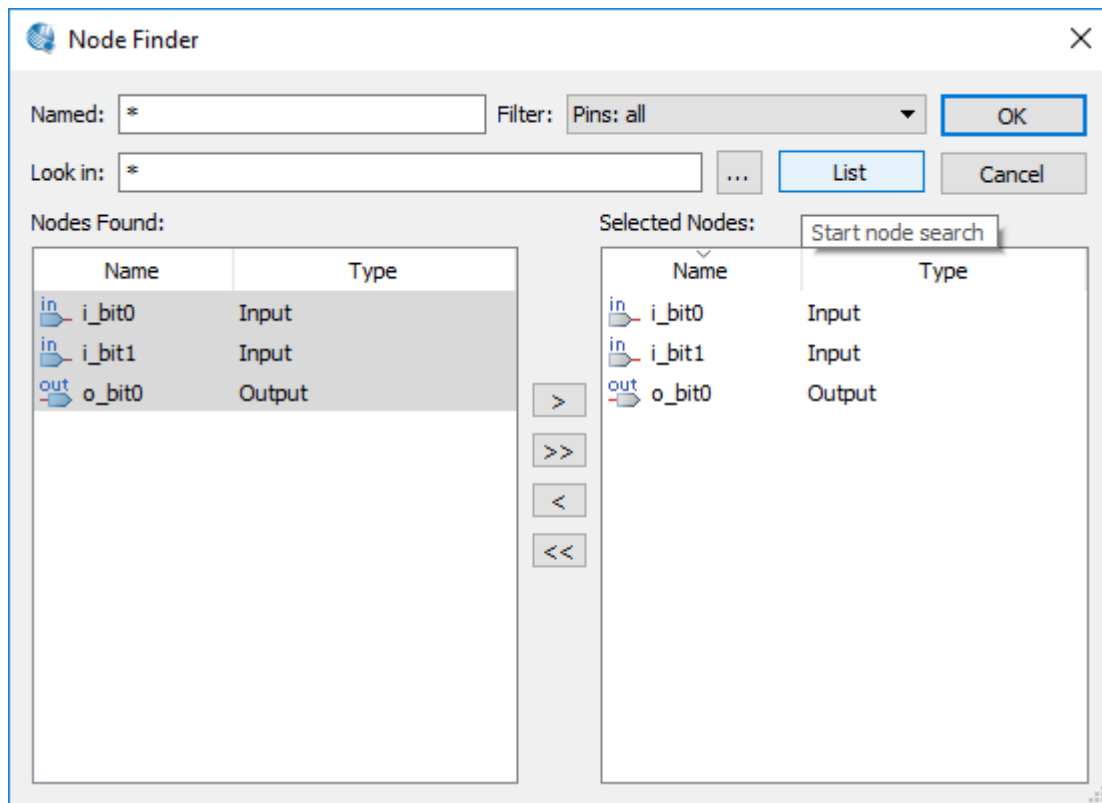
6. Kompiliuojame programos kodą. Sukompiliuotą programą galima įrašyti į FPGA mikrovaldiklį arba patikrinti programos veikimą modeliuojant jos įėjimus ir išėjimus. Modeliavimas skirtas norint sužinoti programos veikimą, t.y. kokie signalai bus programos išėjime, prie tam tikrų įėjimo kombinacijų. Tam tikslui sukuriame naują failą „University Program VWF“.



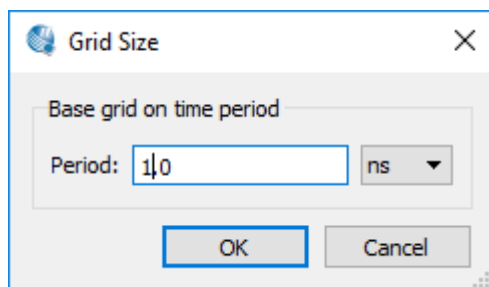
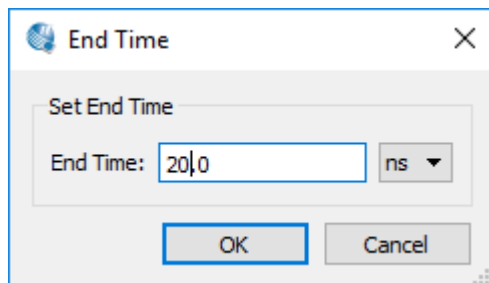
7. Atsidariusiame lange reikia pridėti norimus signalus, pasirenkame Insert Node or Bus, ir spaudžiame „Node Finder...“

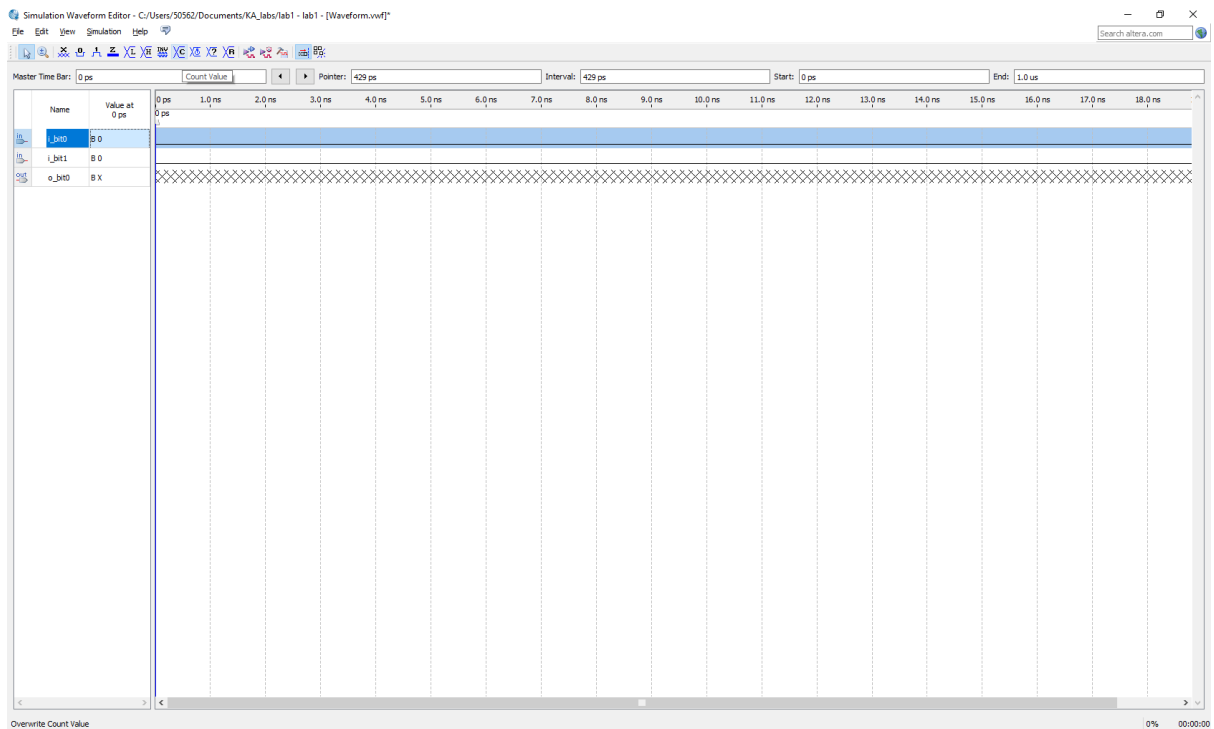


8. Spaudžiame „List“ ir visus rastus mazgus (jų turėtų būti 3) perkeliame į „selected“ dalį.



9. Nustatykite modeliavimo pabaigos laiką ir norimą mastelį tinklui.





10. Įėjimo signalai dabar yra nustatyti į pradinės reikšmės (,0‘). Norint patikrinti ar gerai veikia jūsų aprašytas loginis elementas reikia nustatyti įvairias įėjimo kombinacijas, vienas iš būdų, kuriuo tai galima padaryti yra panaudojant „Count value“ ir nustatant „Transitions occur“ laiką.

Count Value

Radix: **Binary**

Start value: **0**

Increment by: **1**

Count type

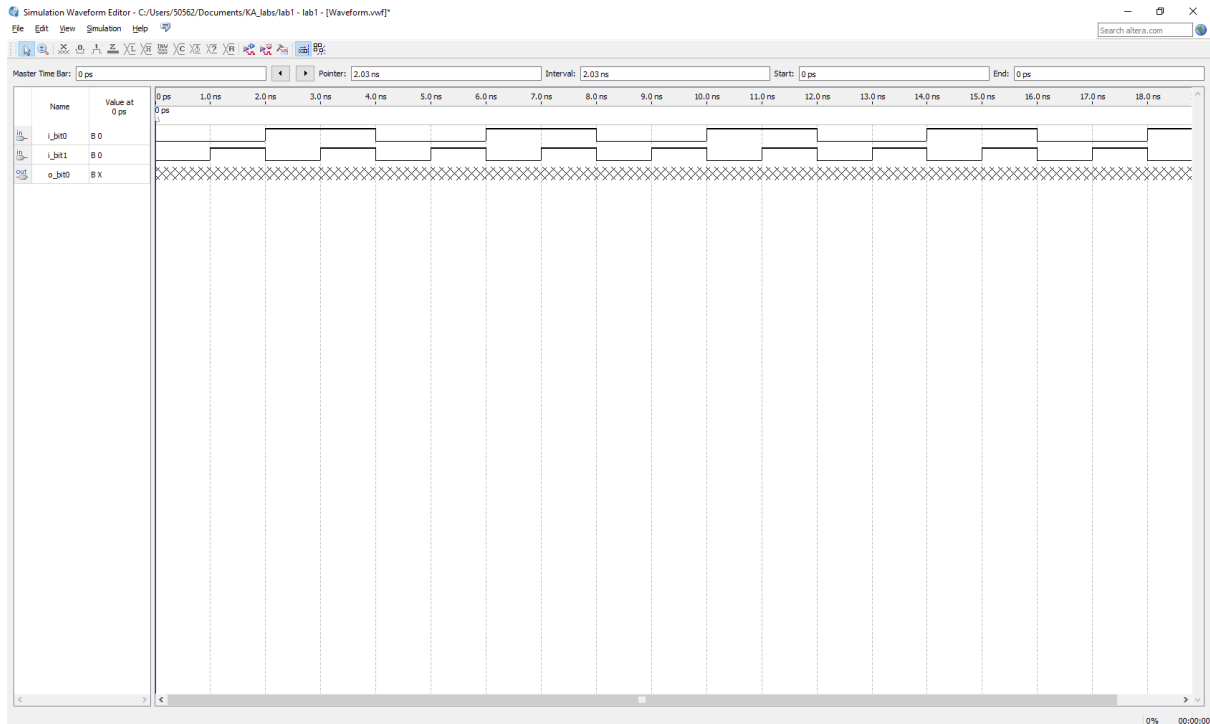
☒ Binary

☐ Gray code

Transitions occur

Count every: **20** ns

OK **Cancel**



11. Spaudžiame „Run functional simulation“¹. Ar geras gavosi rezultatas? Kodėl?

12. Atlikite sekančias užduotis:

- Pakeiskite grandinės aprašytą programos kodą kitu loginiu elementu, ir patikrinkite jo veikimą simuliacija.
- Papildykite programos kodą, taip kad grandinė turėtų ir panaudotų 4 įėjimus ir 1 išėjimą. Parodykite rezultatus simuliacijos rezultatu.
- Papildykite programą, kad grandinė turėtų 2 išėjimus. Parodykite rezultatą.

Kitam laboratoriniam darbui susipažinkite su VHDL programavimo sintakse, perskaitykite mažiausiai pirmus tris skyrius iš knygos „Free Range VHDL“, kuri yra GitHub repozitorijoje:

<https://github.com/VytAbr/KA-l.d.>

Hints:

1. Jei neveikia simuliacija – Tools > options > EDA tool options > ModelSim-Altera > katalogą pakeisti į „modelsim_ase“.

