

**Informatikos fakultetas**

**SKAITMENINĖS LOGIKOS PRADMENYS**

**Individualios užduoties Nr. 109**

**Atliko:** Vytenis Kriščiūnas gr. Stud. IFF-1/1

**Primėmė:** dėst. Stasys Maciulevičius

**Kaunas, 2022**

TURINYS

1. ILIUSTRACIJŲ SĄRAŠAS 3

2. ĮVADAS 4

3. NAGRINĖJAMOS TEORINĖS DARBO PRIELAIDOS 5

4. INDIVIDUALIOS UŽDUOTIES PROJEKTAVIMO ETAPAI 7

4.1 Skaitiliai. 7

4.2 M1 skaitiklio VHDL kodas. 7

4.3 M2 skaitiklio VHDL kodas. 7

4.4 M3 skaitiklio VHDl kodas. 8

4.5 JM1 skaitiklio nustatymo į nulinę būseną sąlygos skaičiavimas. 9

4.6 JM1 skaitiklio VHDL kodas. 9

4.7 JM2 nustatymo į nulinę būseną sąlygos skaičiavimas. 10

4.8 JM2 skaitiklio VHDL kodas. 11

4.9 ModelSim testai. 12

4.10 ModelSim programinės įrangos gauti rezultatai. 13

4.11 RTL hierarchinė realizalicja. 14

5. IŠVADOS 16

1. ILIUSTRACIJŲ SĄRAŠAS

1 pav. Trigerių skaičiaus radimo formulė 5

2 pav. Asichroninis skaitiklis 5

3 pav. JM1 skaitiklio vidinio perkrovimo sąlygos skaičiavimo formulė 6

4 pav. JM2 skaitiklio vidinio perkrovimo sąlygos skaičiavimo formulė 6

5 pav. skaitikliai 7

6 pav. M1 skaitiklio VHDL kodas 7

7 pav. M2 skaitiklio VHDL kodas 8

8 pav. M3 skaitiklio VHDL kodas 9

9 pav. JM1 nustatymo į nulinę būseną sąlygos skaičiavimas 9

10 pav. JM1 skaitiklio VHDL kodas 10

11 pav. JM2 nustatymo į nulinę būseną sąlygos skaičiavimas 11

12 pav. JM2 skaitiklio VHDL kodas 12

13 pav. Testavimo direktyvos 13

14 pav. M1 skaitiklio ModelSim rezultatai 13

15 pav. M2 skaitiklio ModelSim rezultatai 13

16 pav. M3 skaitiklio ModelSim rezultatai 14

17 pav. JM1 skaitiklio ModelSim rezultatai 14

18 pav. JM2 skaitiklio ModelSim rezultatai 14

19 pav. Symplify Pro JM1 skaitiklio schema 14

20 pav. Simplify Pro JM2 skaitiklio schema 15

1. ĮVADAS

Šio individualaus darbo tikslas buvo išmokti įvairių skaitiklių ir daliklių veikimo principus, suprasti jų realizavimą VHDL kalboje, įsigilinti į projektavimo ir taikymo galimybes. Atliekant užduotis patikrinti skaitiklių ir daliklių veikimą simuliacijoje ir programuojamos logikos schemose. Remiantis pateiktais pavyzdžiais teorinėje medžiagoje pilnai atlikti priskirtą individualią užduotį.

1. NAGRINĖJAMOS TEORINĖS DARBO PRIELAIDOS

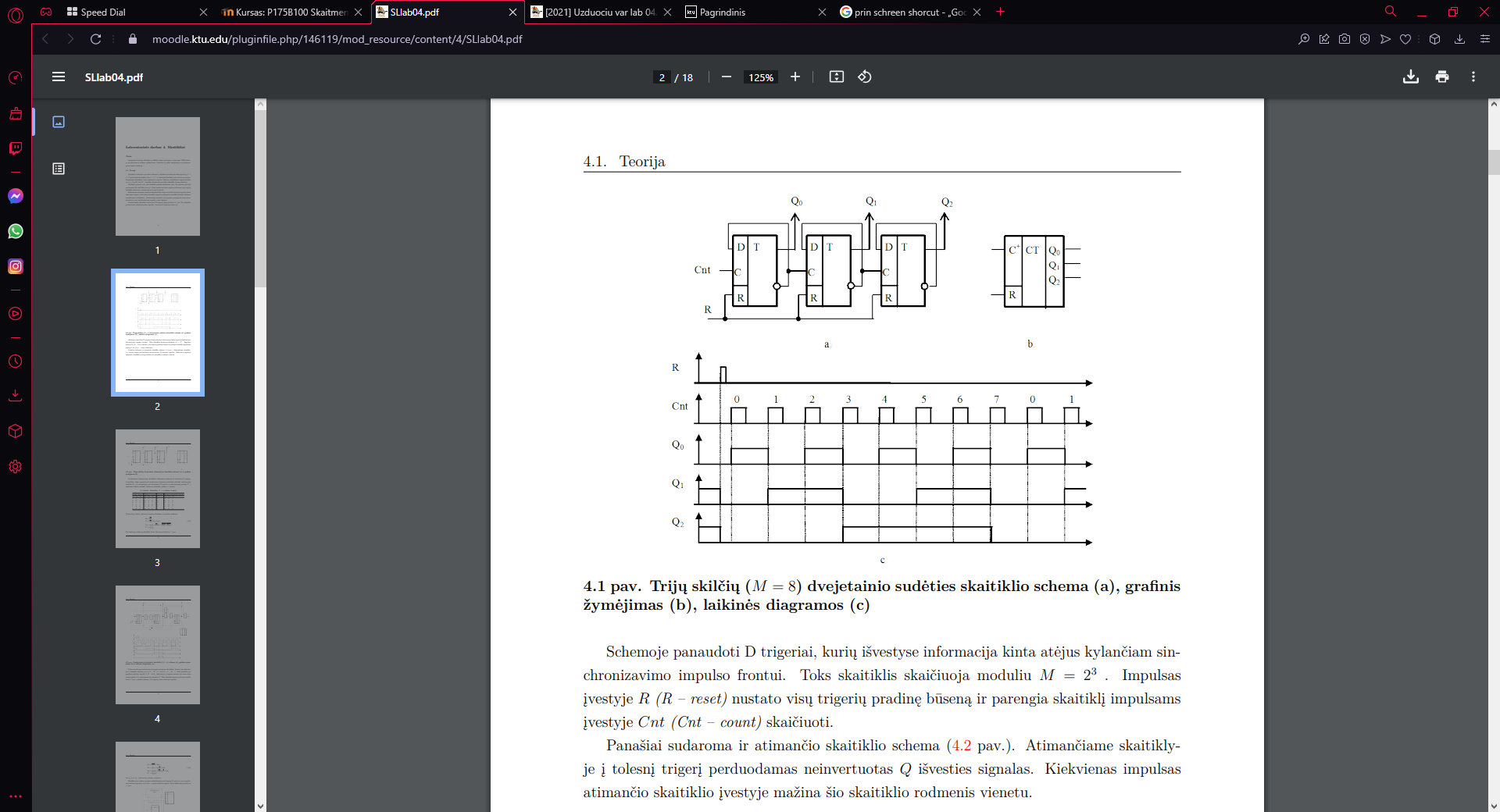
Siekdamas atlikti individualią užduotį turėjau išnagrinėti teorinę šio darbo dalį. Teorinėje medžiagoje išsisaiškinau, kad skaitikliais yra vadinami operaciniai elementai, atliekantys skaičiavimo mikrooperacijas, pavyzdžiui: pridedantys skaitikliai, atimantys skaitikliai arba abi šias operacijas. Skaitikliai, kurios teko nagrinėti buvo sudaromi iš trigerių. O pats trigerių skaičius buvo nagrinėjamas pagal formulę:



1 pav. Trigerių skaičiaus radimo formulė

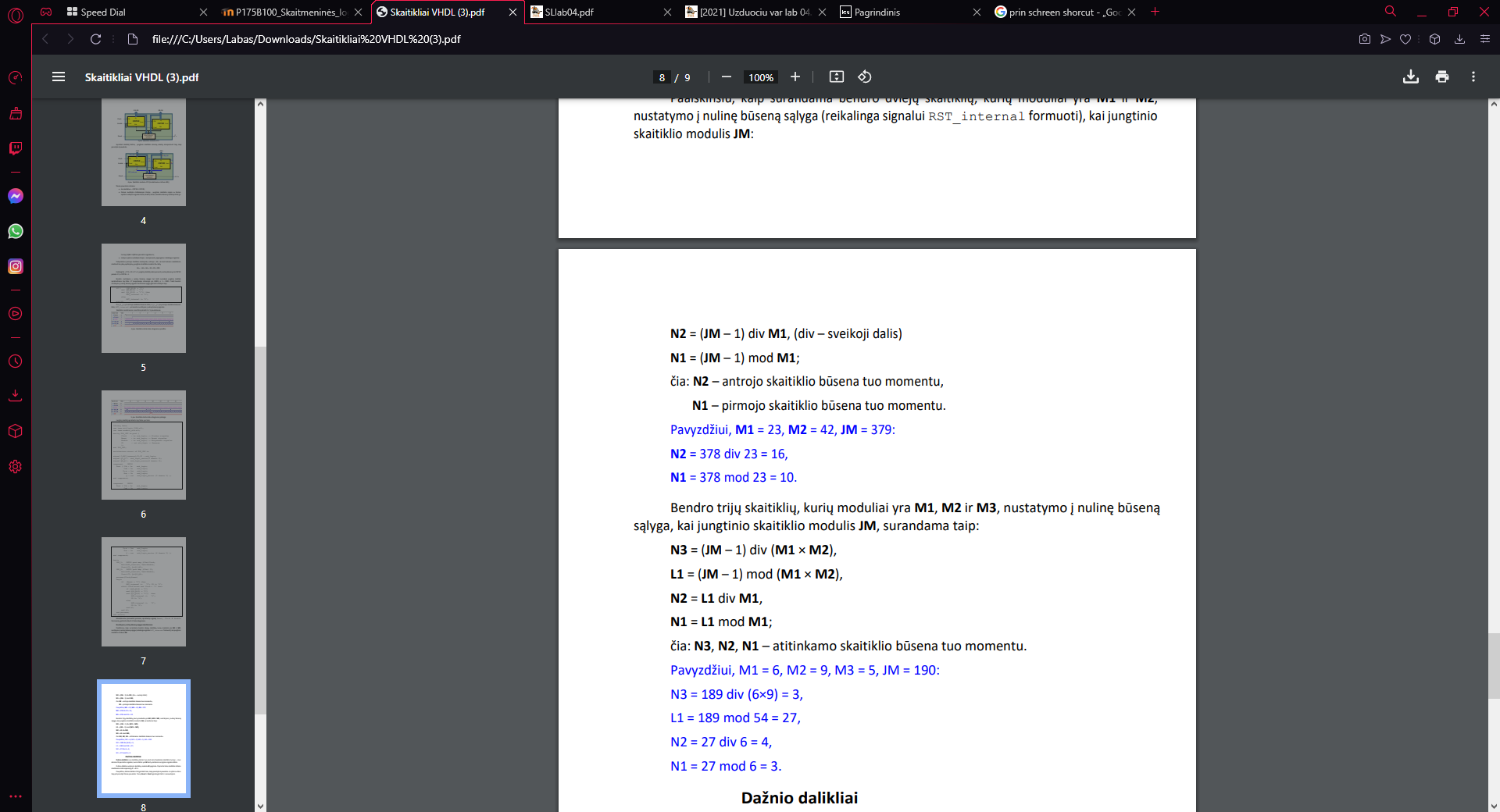
Skaitikliui pasiekus skaičiavimo ribą, jis suformuoja pernašą, kurią gali naudoti kitas skaitiklis. Trigerių skaitikliai yra skirstomi į asinchroninius ir sinchroninius. Sinchroninių skaitiklių visi trigeriai būna perjungiami tuo pačiu metu, kai yra išsiunčiamas tas pats sinchronizavimo signalas.

Asinchroniniai skaitikliai susidaro iš D trigerių, kaip parodyta apačioje esančiame paveiklėlyje.

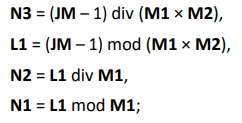


2 pav. Asichroninis skaitiklis

Individualioje užduotyje pateiktus skaitiklius reikėjo realizuoti VHDL kalboje. Pagal pateiktas formules 3 ir 4 paveiklėliuose buvo galima apskaičiuoti JM1 ir JM2 skaitiklių nustatymo į nulinę būseną sąlygas.



3 pav. JM1 skaitiklio vidinio perkrovimo sąlygos skaičiavimo formulė



4 pav. JM2 skaitiklio vidinio perkrovimo sąlygos skaičiavimo formulė

1. INDIVIDUALIOS UŽDUOTIES PROJEKTAVIMO ETAPAI
   1. Skaitiliai.

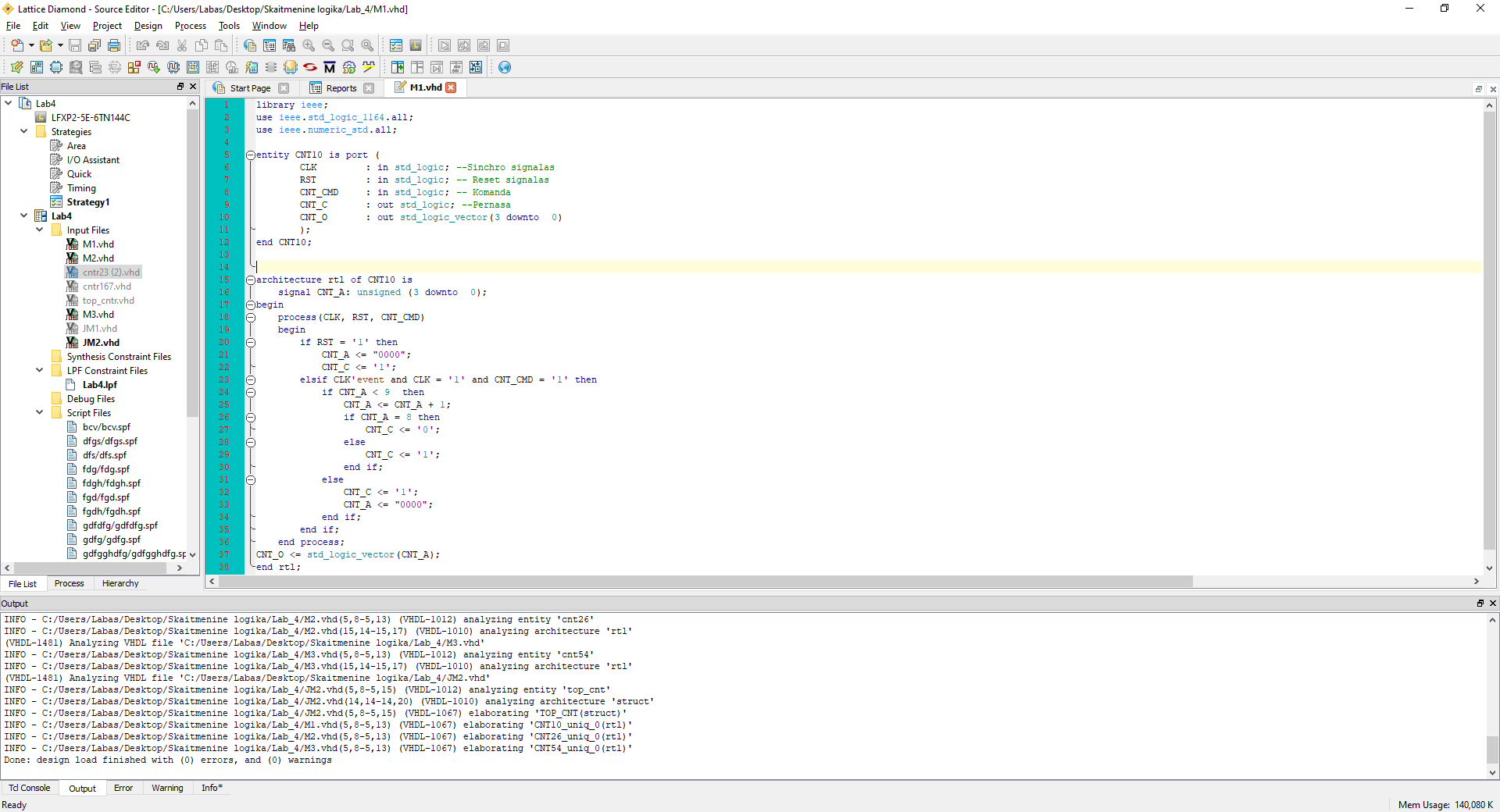
Nagrinėjau man priskirtos individualios užduoties skaitiklius, remiantis pateikta teorija.



5 pav. skaitikliai

* 1. M1 skaitiklio VHDL kodas.

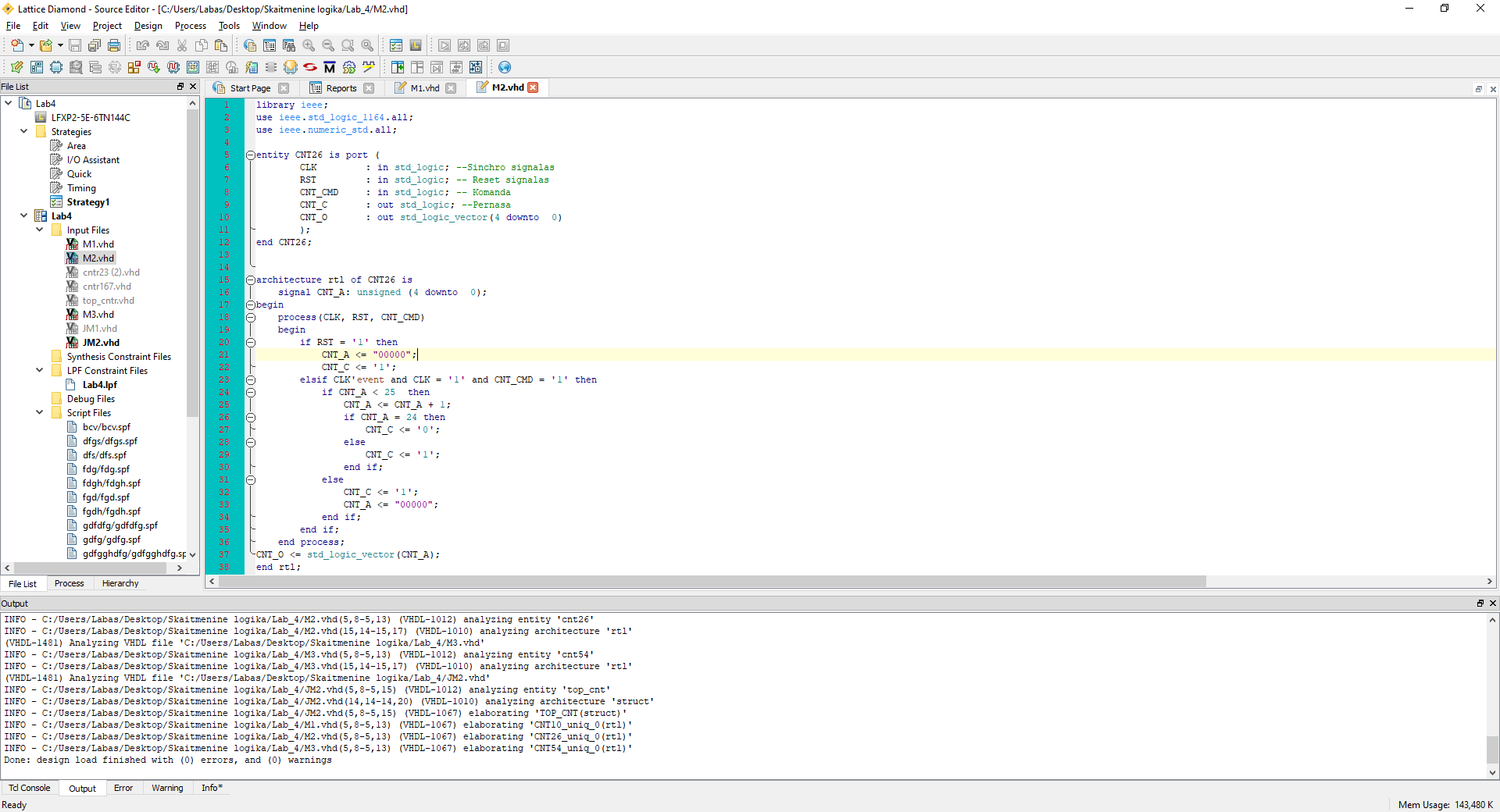
Pakeičiau teorinėje mežiagoje pateiktą M1 VHDL kodą, kad atitiktų man priskirtoje salygoje M1 skaitiklį.



6 pav. M1 skaitiklio VHDL kodas

* 1. M2 skaitiklio VHDL kodas.

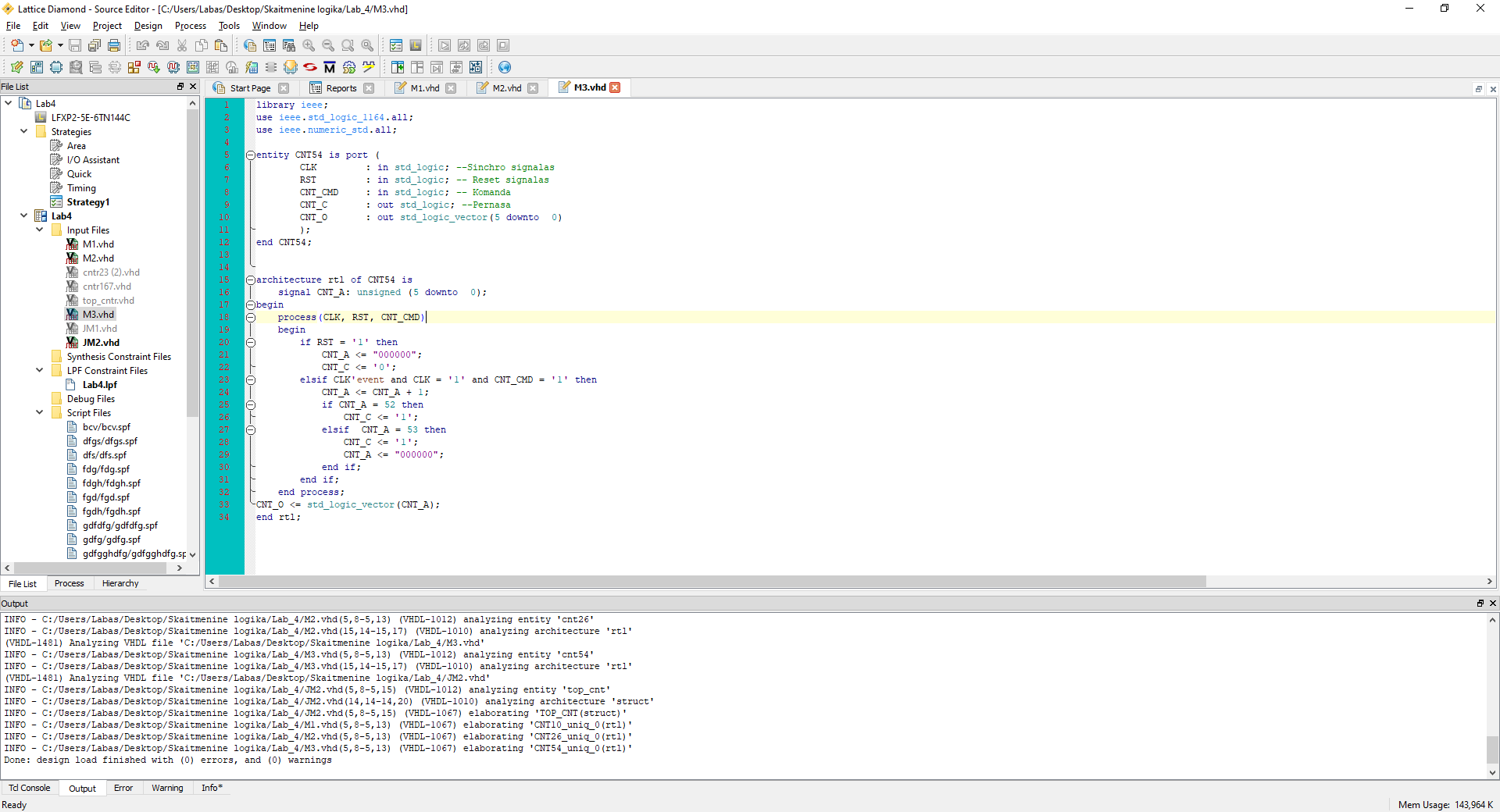
Pakeičiau teorinėje medžiagoje pateiktą M2 VHDL kodą, kad atitiktų man priskirtoje sąlygoje M2 skaitiklį.



7 pav. M2 skaitiklio VHDL kodas

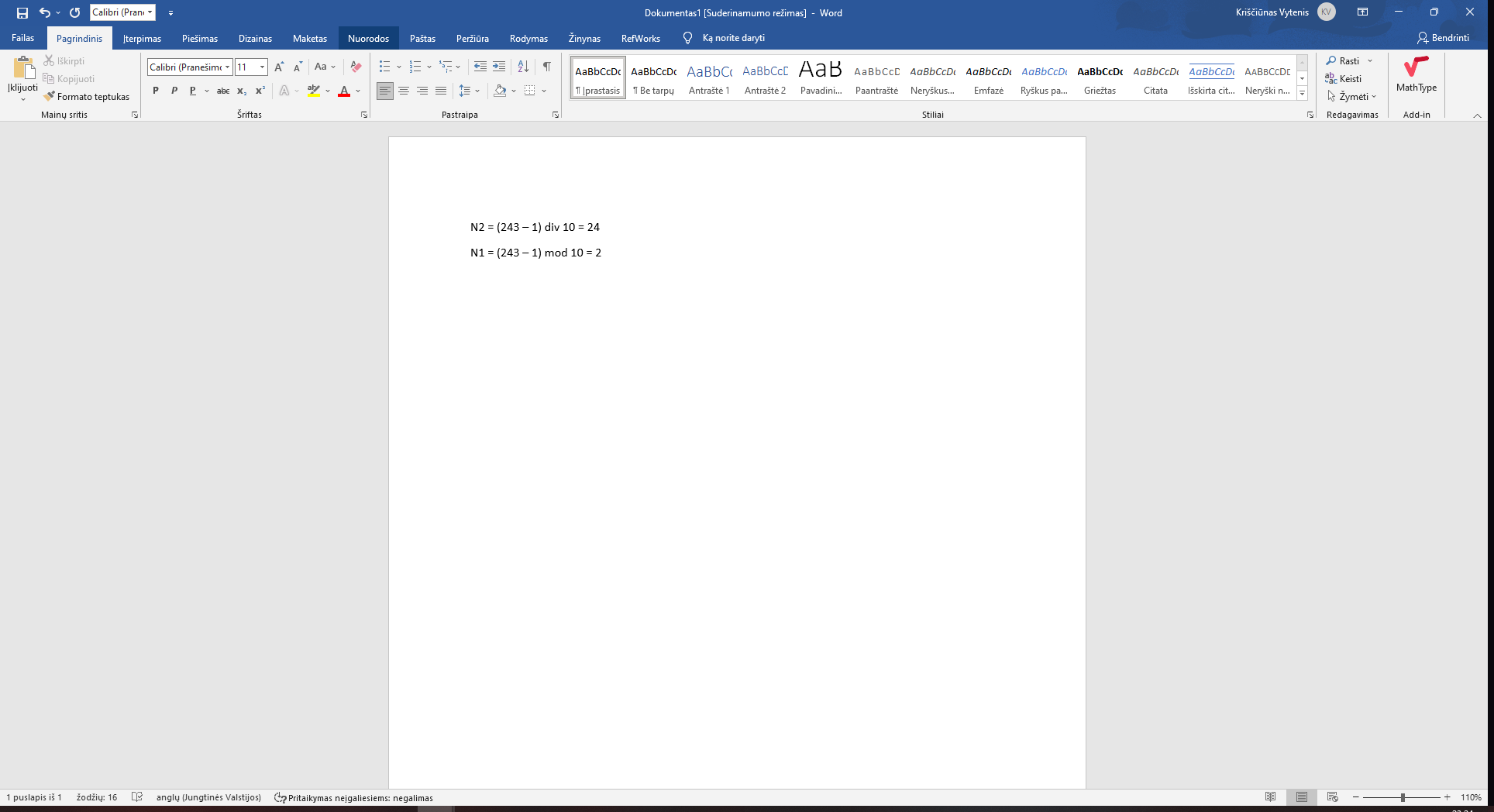
* 1. M3 skaitiklio VHDl kodas.

Pakeičiau teorinėje medžiagoje pateiktą M3 VHDL kodą, kad atitiktų man priskirtoje sąlygoje M3 skaitiklį.



8 pav. M3 skaitiklio VHDL kodas

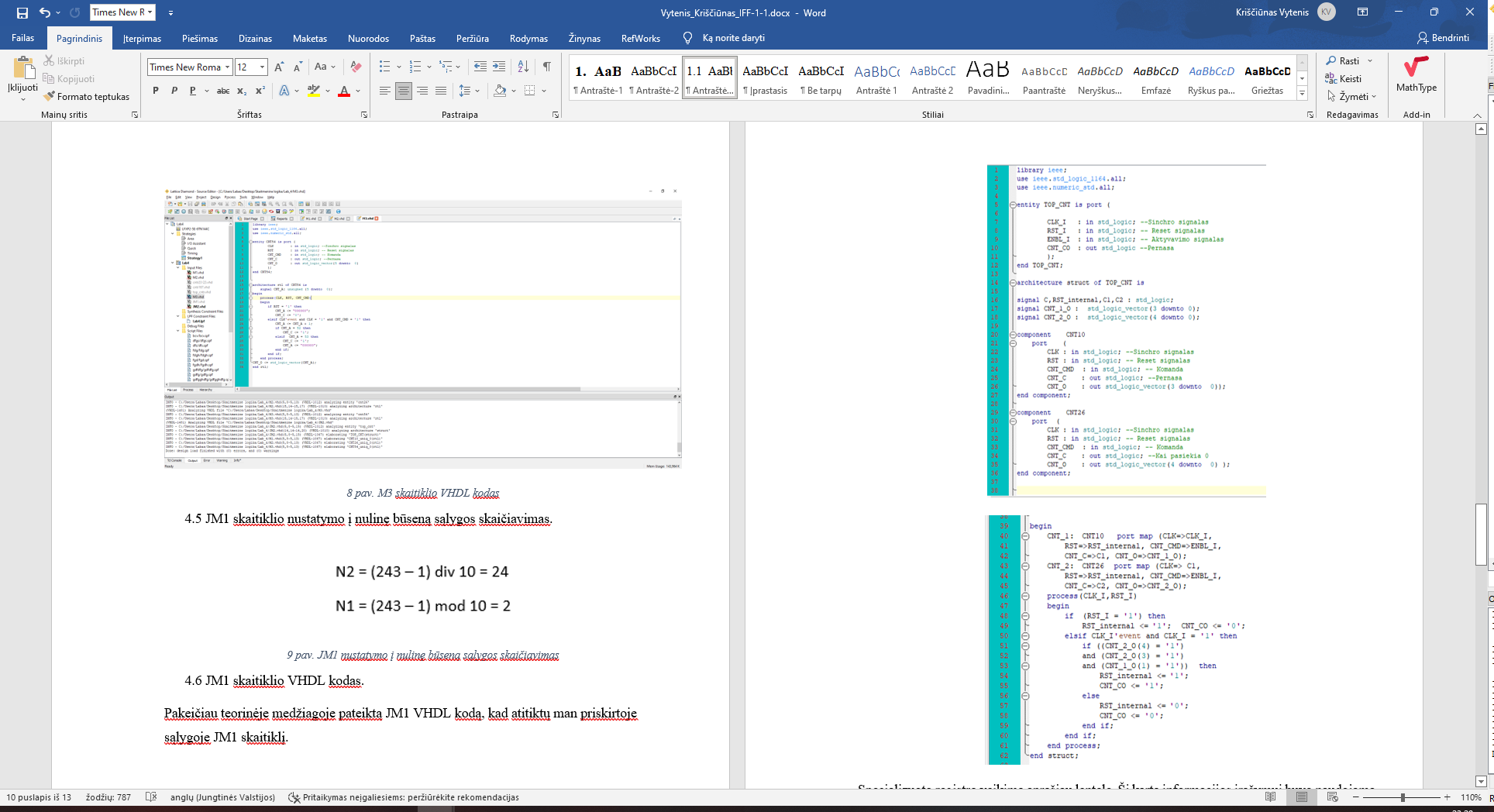
* 1. JM1 skaitiklio nustatymo į nulinę būseną sąlygos skaičiavimas.



9 pav. JM1 nustatymo į nulinę būseną sąlygos skaičiavimas

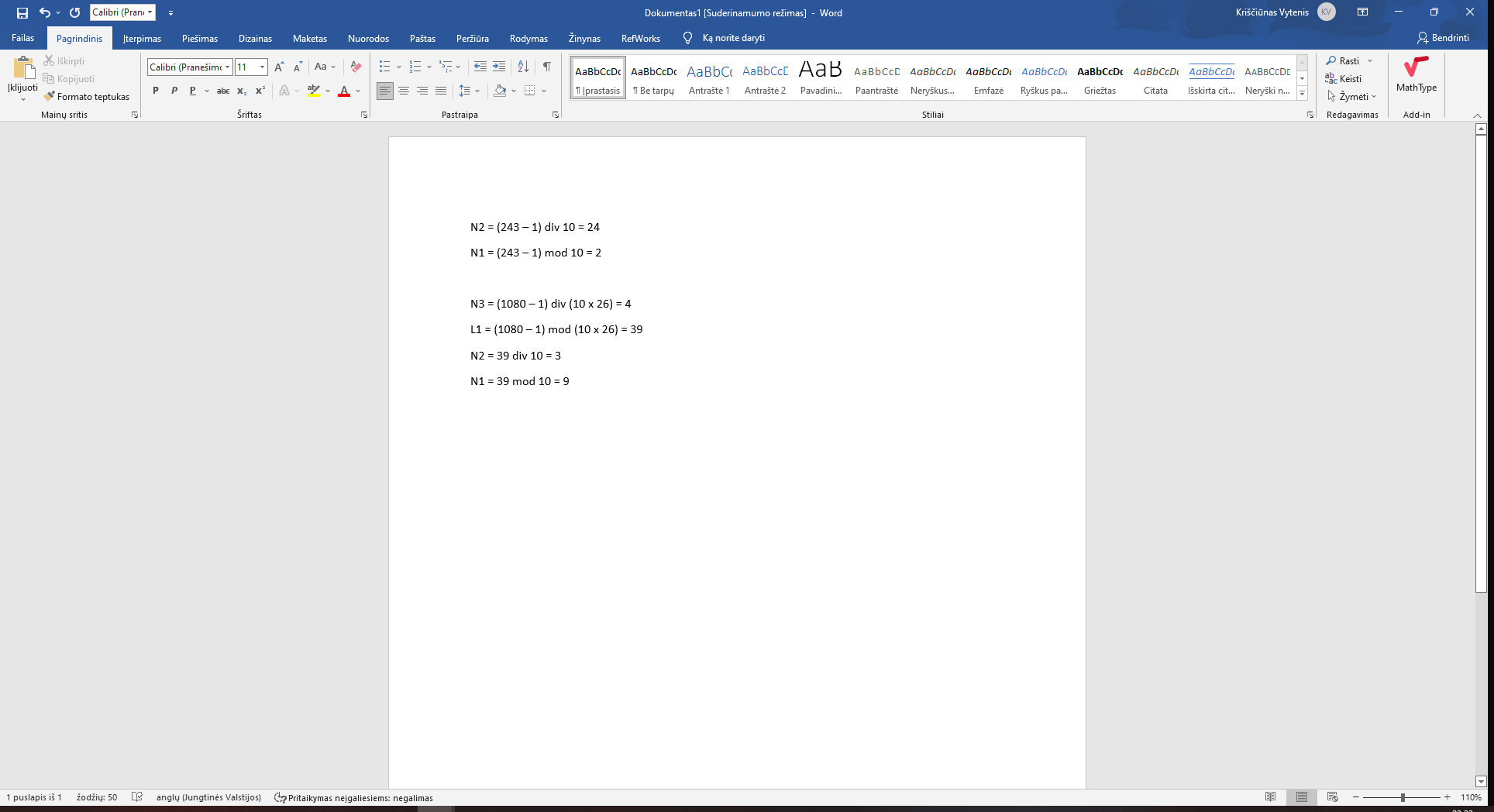
* 1. JM1 skaitiklio VHDL kodas.

Pakeičiau teorinėje medžiagoje pateiktą JM1 VHDL kodą, kad atitiktų man priskirtoje sąlygoje JM1 skaitiklį.



10 pav. JM1 skaitiklio VHDL kodas

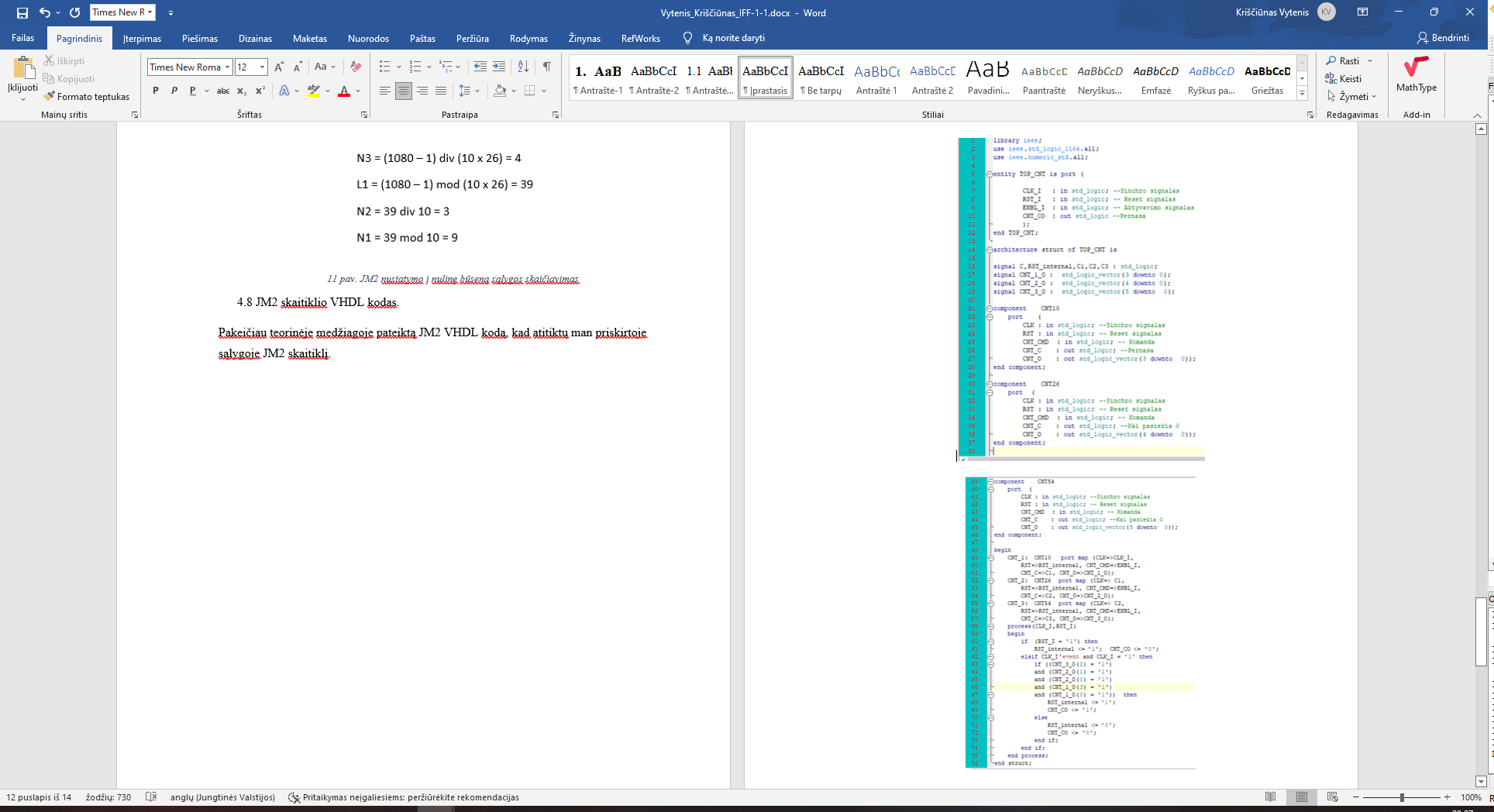
* 1. JM2 nustatymo į nulinę būseną sąlygos skaičiavimas.



11 pav. JM2 nustatymo į nulinę būseną sąlygos skaičiavimas

* 1. JM2 skaitiklio VHDL kodas.

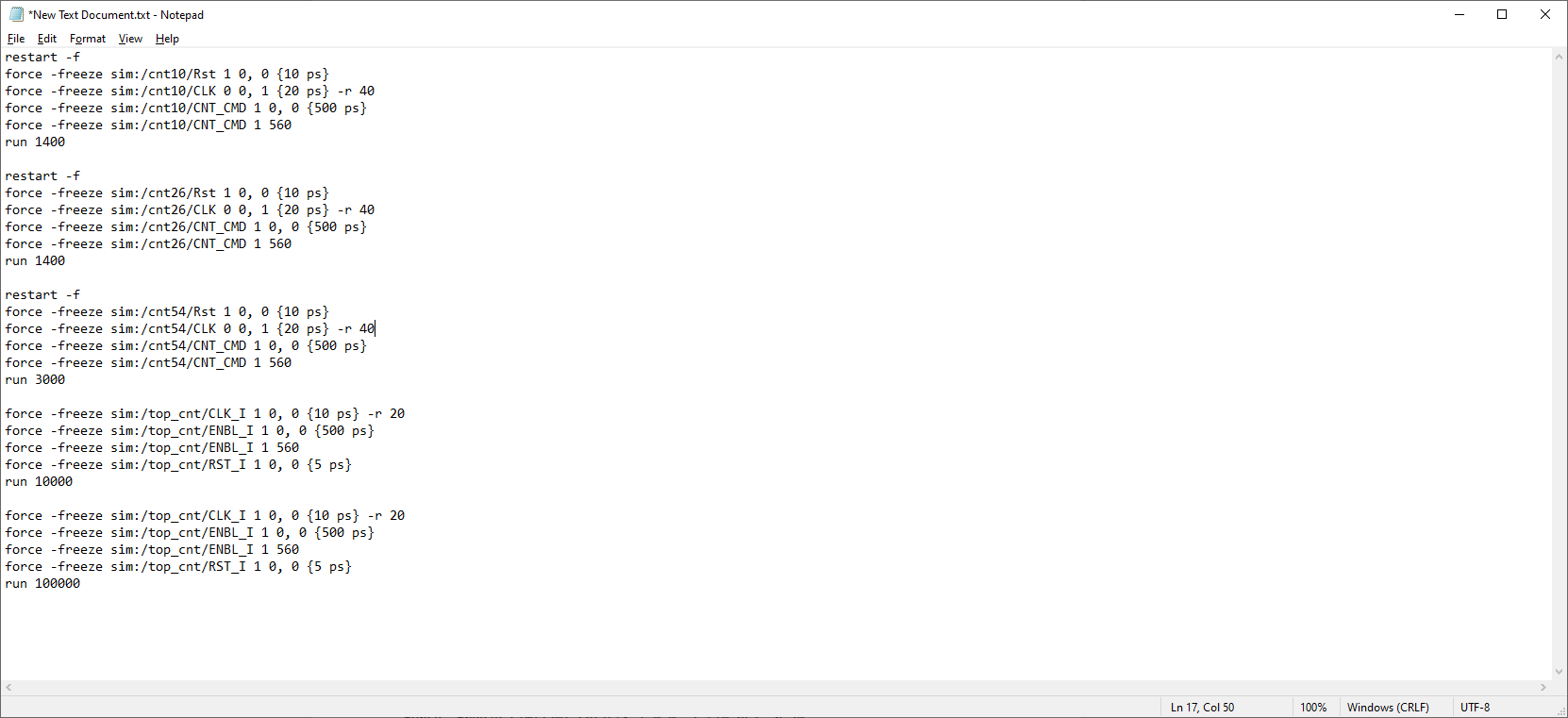
Pakeičiau teorinėje medžiagoje pateiktą JM2 VHDL kodą, kad atitiktų man priskirtoje sąlygoje JM2 skaitiklį.



12 pav. JM2 skaitiklio VHDL kodas

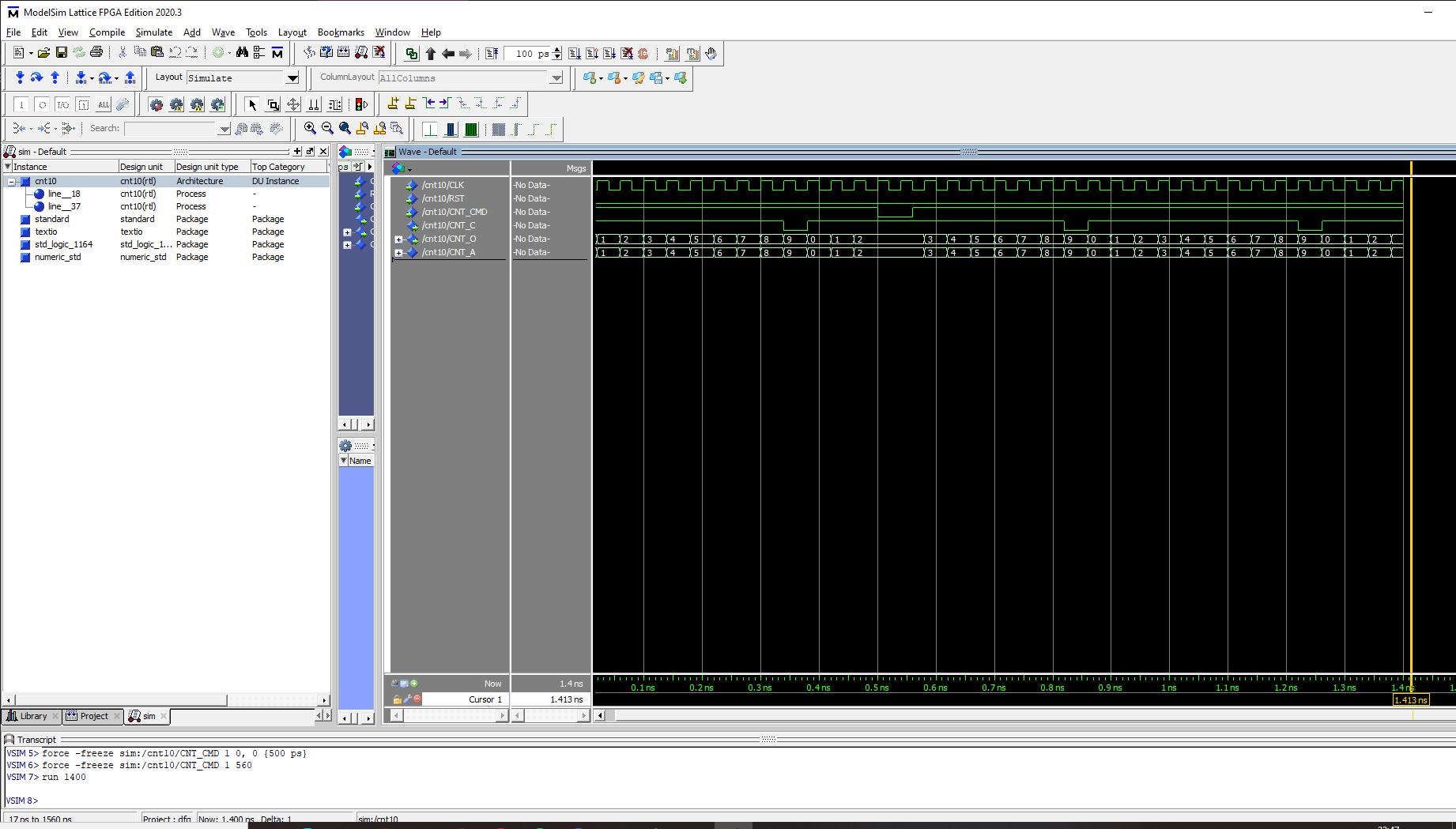
* 1. ModelSim testai.

Susidariau testavimo direktyvas.

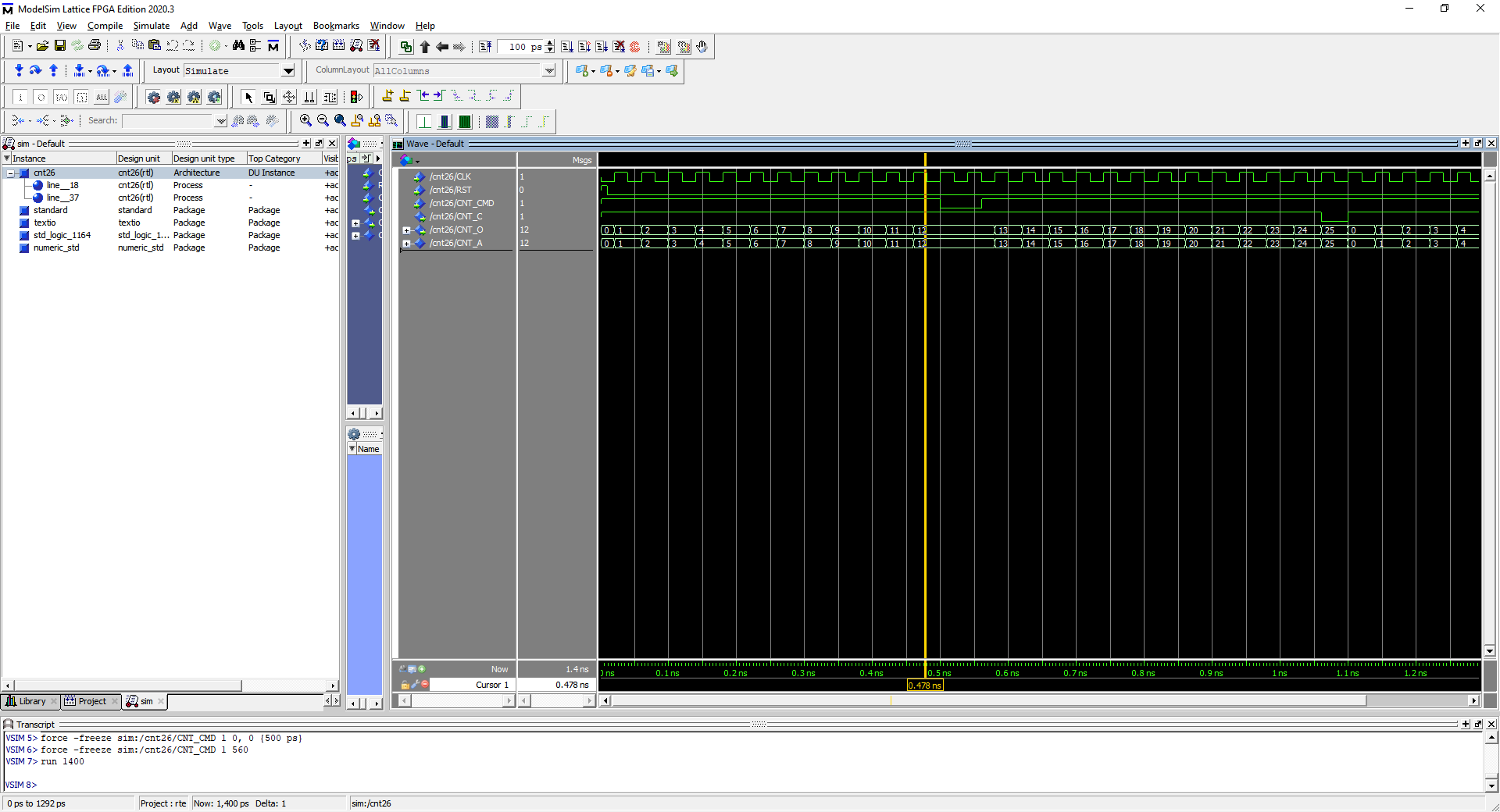


13 pav. Testavimo direktyvos

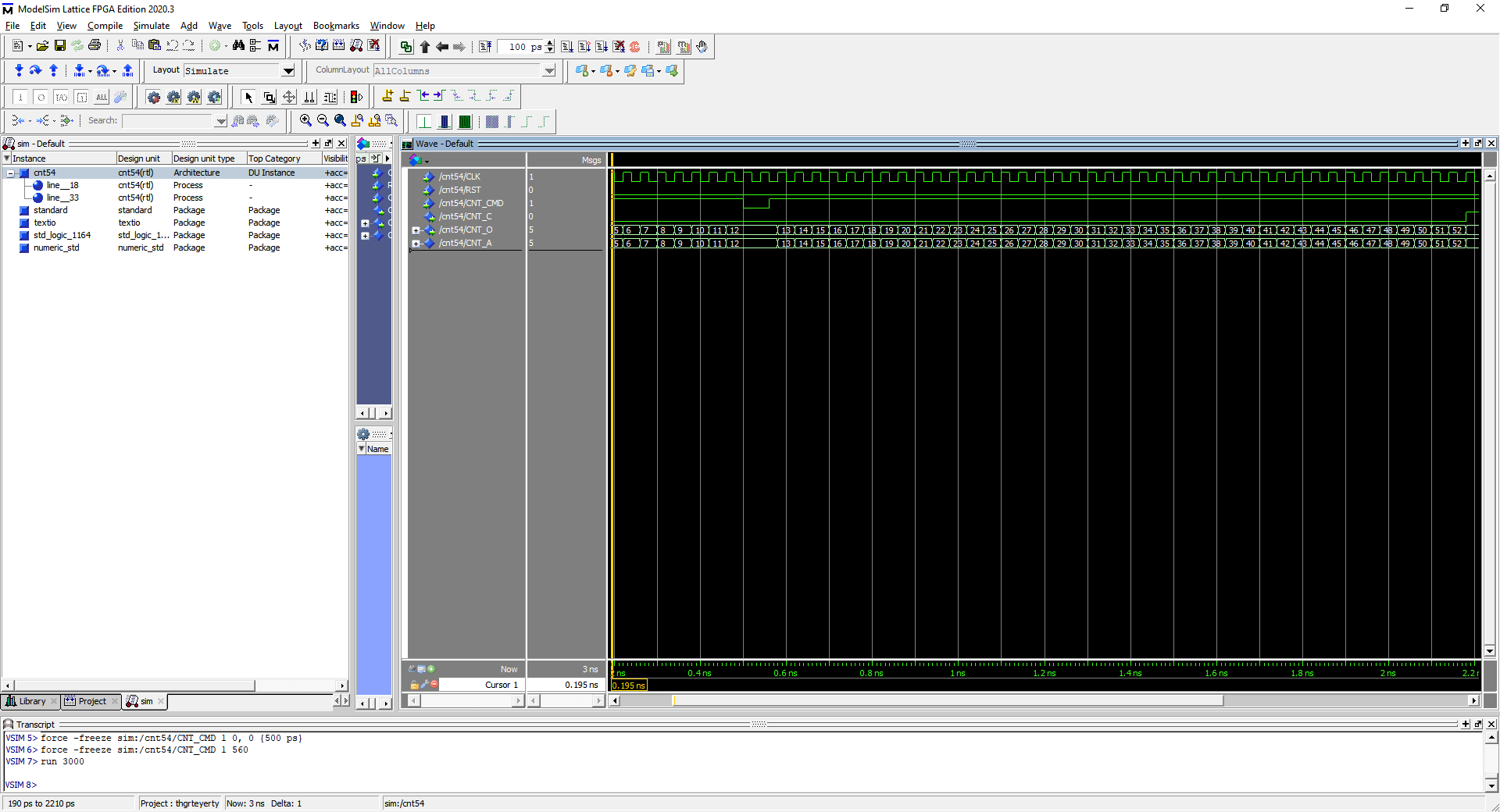
* 1. ModelSim programinės įrangos gauti rezultatai.



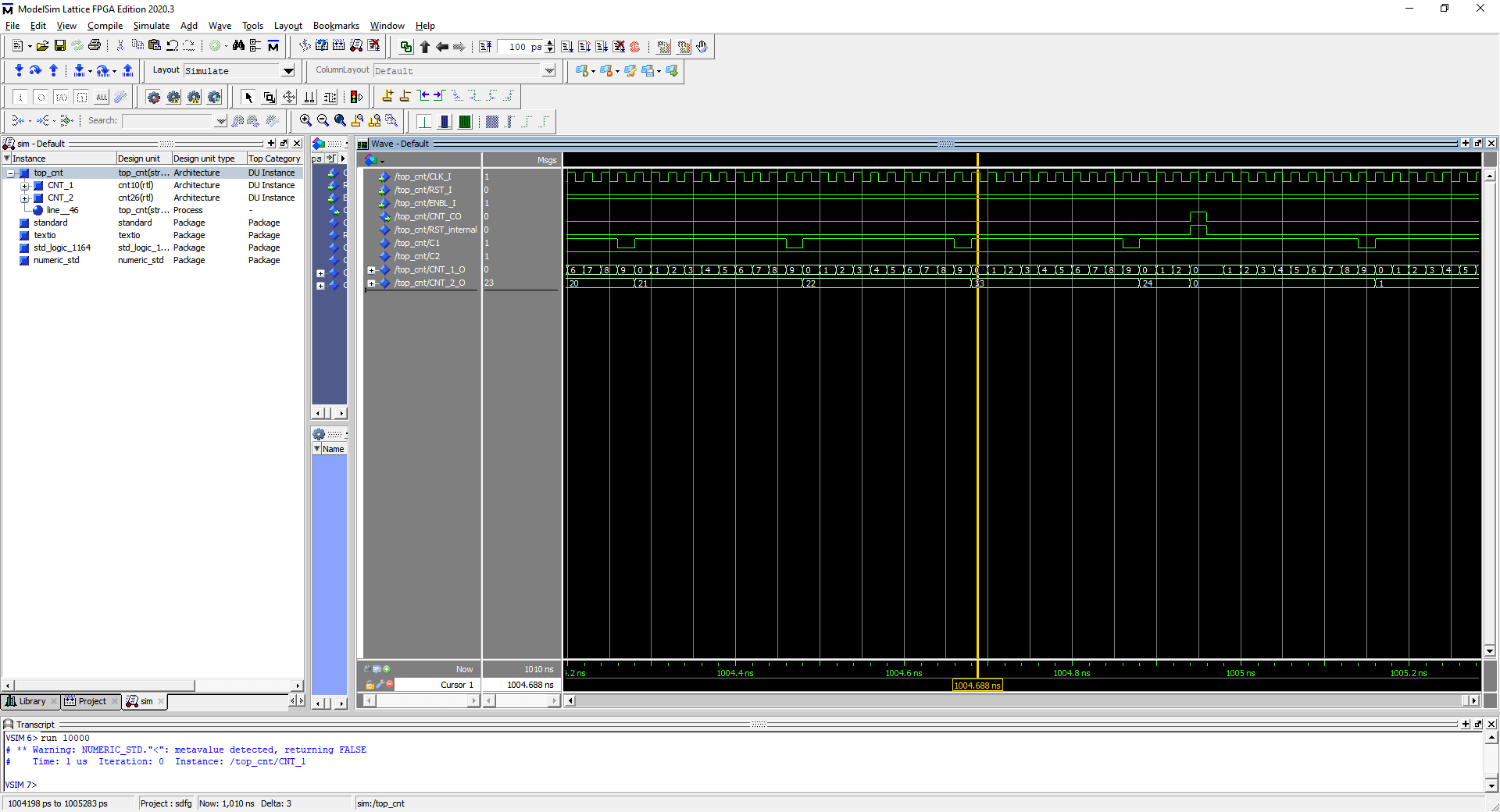
14 pav. M1 skaitiklio ModelSim rezultatai



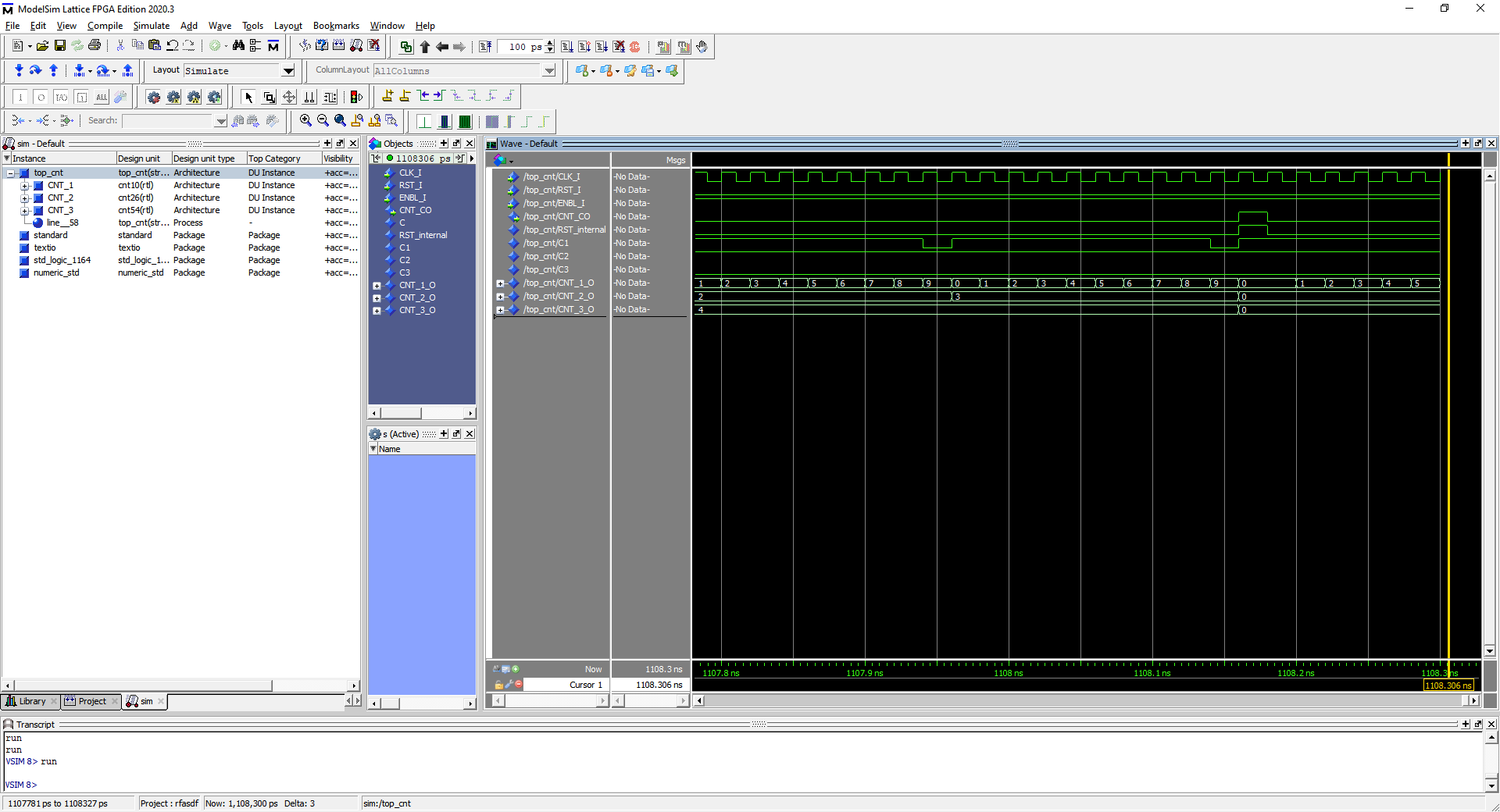
15 pav. M2 skaitiklio ModelSim rezultatai



16 pav. M3 skaitiklio ModelSim rezultatai

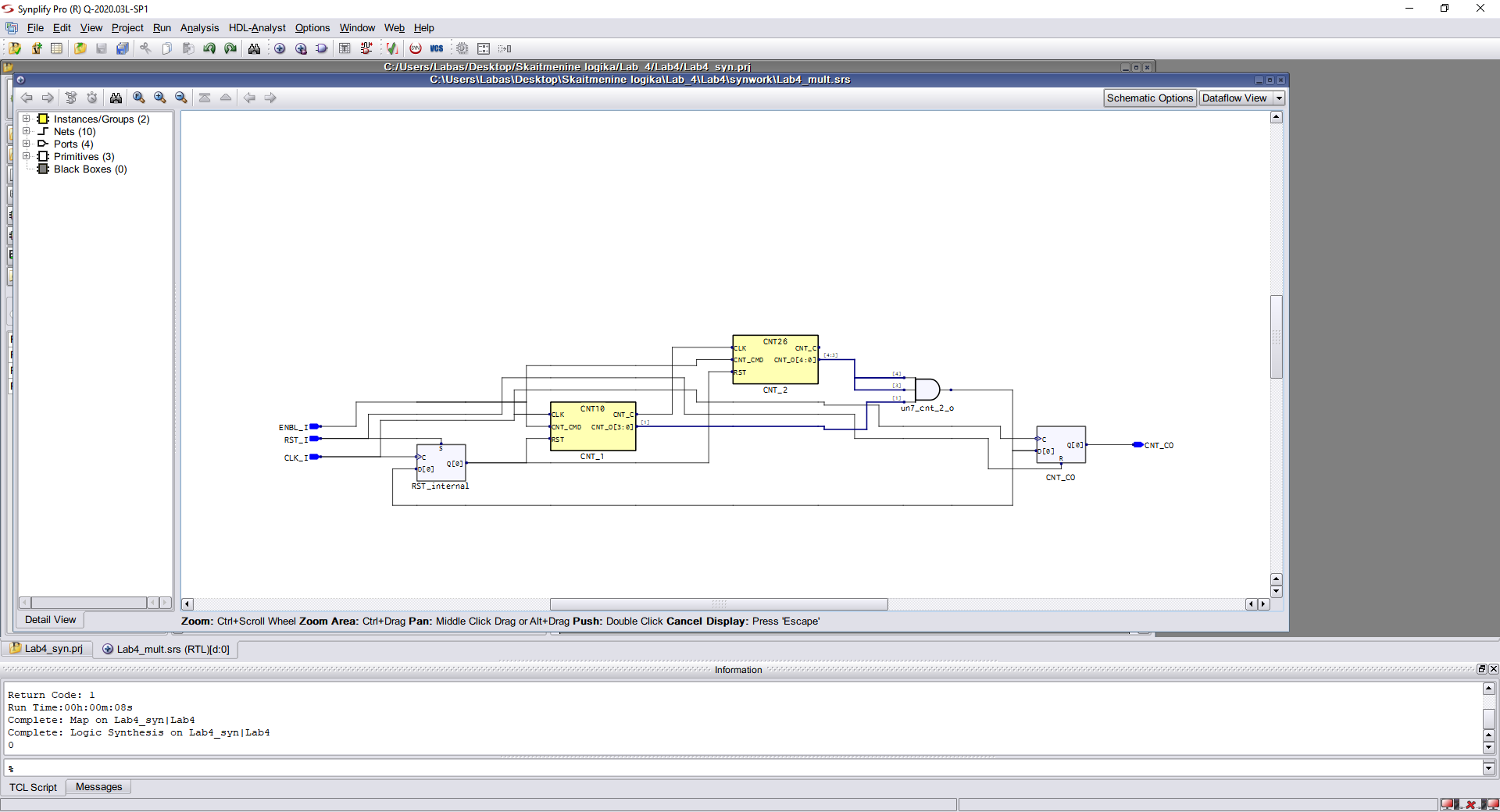


17 pav. JM1 skaitiklio ModelSim rezultatai

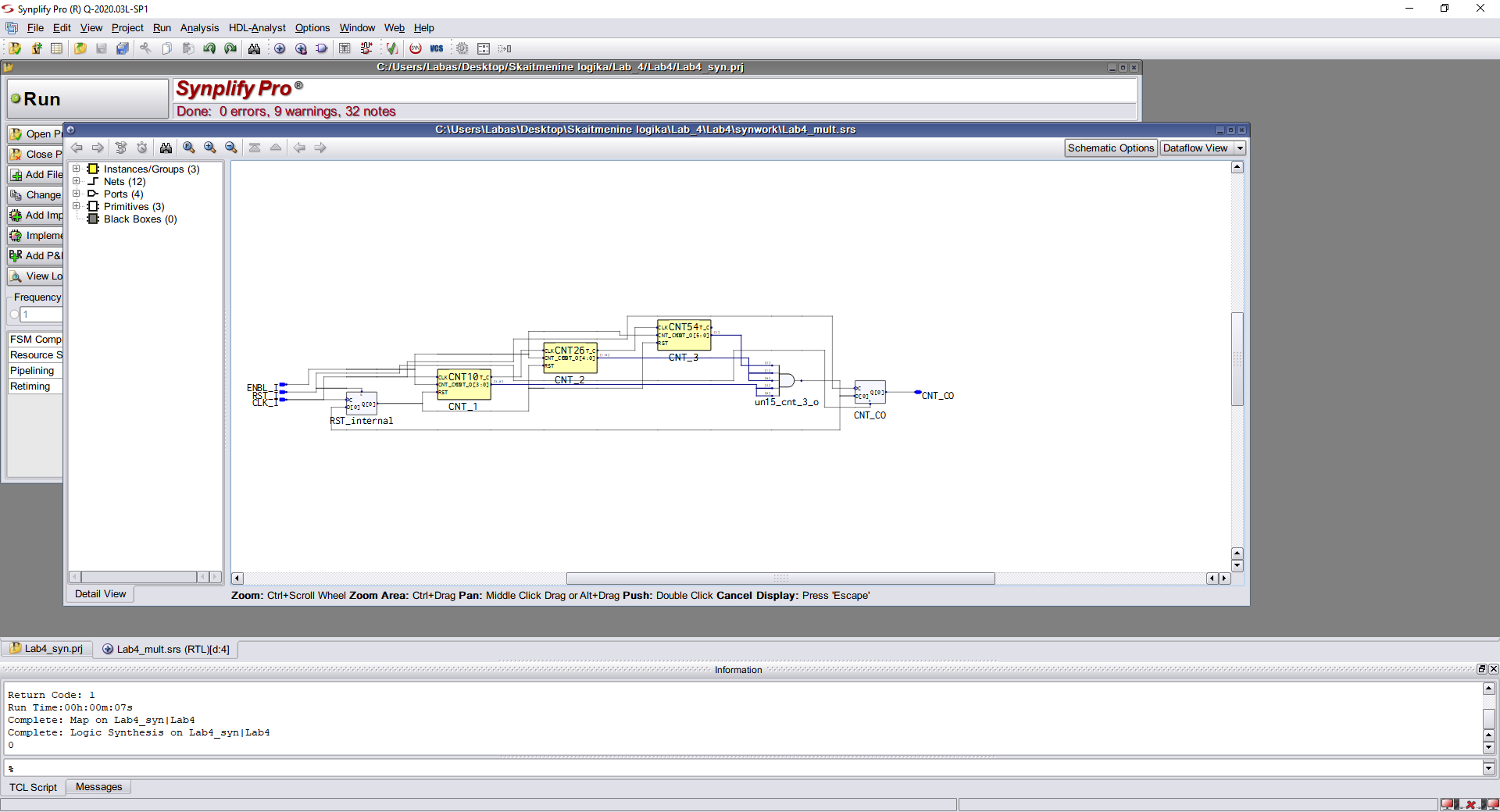


18 pav. JM2 skaitiklio ModelSim rezultatai

* 1. RTL hierarchinė realizalicja.



19 pav. Symplify Pro JM1 skaitiklio schema



20 pav. Simplify Pro JM2 skaitiklio schema

1. IŠVADOS

Ši individuali užduotis sukėlė man įvairiausių iššūkių, kuriuos pavyko įveikti pasiskaičius pateiktą teorinę paskaitų medžiagą. Sunkiausia darbo dalis buvo atlikti nustatymo į nulinę būseną sąlygos skaičiavimus, pasitelkus duotas formules. Patikrinti skaitiklius ModelSim programoje, naudojant sudarytas testines direktyvas, sunku nebuvo. Daugiausiai laiko skyriau teorinės darbo dalies supratimui. Pateikta informacija buvo labai naudinga sudarinėjant VHDL kodus. Įgytos žinios privedė prie individualios užduoties realizacijos.