

**Универзитет у Београду
Електротехнички факултет
Катедра за електронику**

**Основи дигиталне електронике
- 13E042ОД -**

Домаћи задатак

**Пројектовање хардвера коришћењем
VHDL језика
Контролер семафора**

Београд, јун 2026.

1. Увод

Циљ овог домаћег задатка је да се студенти кроз реализацију контролера семафора на раскрсници упознају са основним концептима пројектовања хардвера у *VHDL* језику. За симулацију *VHDL* кода користи се алат *ModelSim-Intel FPGA*, чије је упутство за инсталацију доступно на следећем линку:

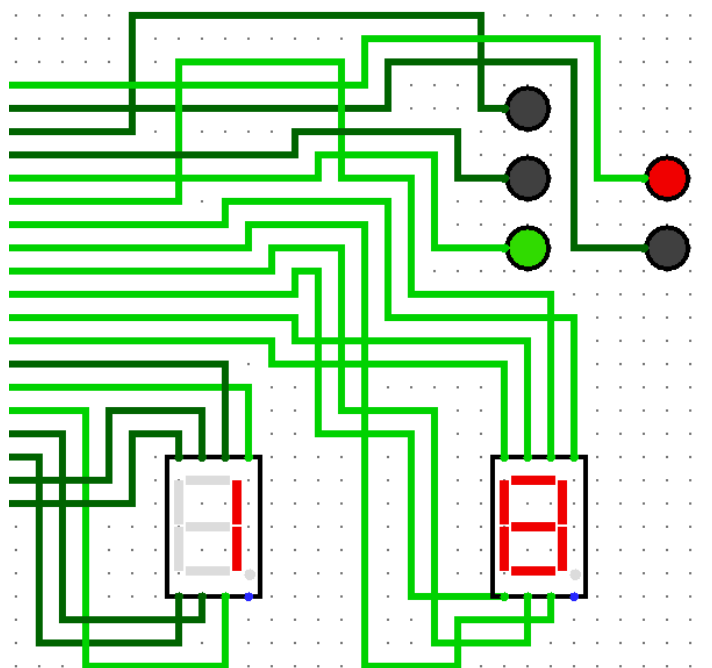
http://tnt.etf.bg.ac.rs/~of2ode/materijali/vezbe/05_vhdl/modelsiminstall.pdf.

За визуелизацију рада пројектованог система користи се алат *Logisim* чија је инсталација доступна на следећем линку:

<https://sourceforge.net/projects/circuit/>. Инсталација ће потенцијално затражити додатну инсталацију Java Development Kit-a, па је потребно испратити тражену процедуру.

У оквиру домаћег задатка потребно је да студенти креирају неколико комбинационих кола и једноставну машину стања у *VHDL* језику за опис хардвера. Сваки од модула је потребно симулирати и на тај начин верификовати исправност рада.

На крају се излазни сигнали симулираног дизајна доводе на симулатор семафора са светлима за возаче и пешаке и са *LED* дисплејом за приказ преосталог времена током кога је укључено зелено светло за возаче. Изглед симулатора у кога је потребно учитати излазне сигнале је приказан на слици 1. Нацртане линије представљају излазе модула који је потребно пројектовати у *VHDL*-у.



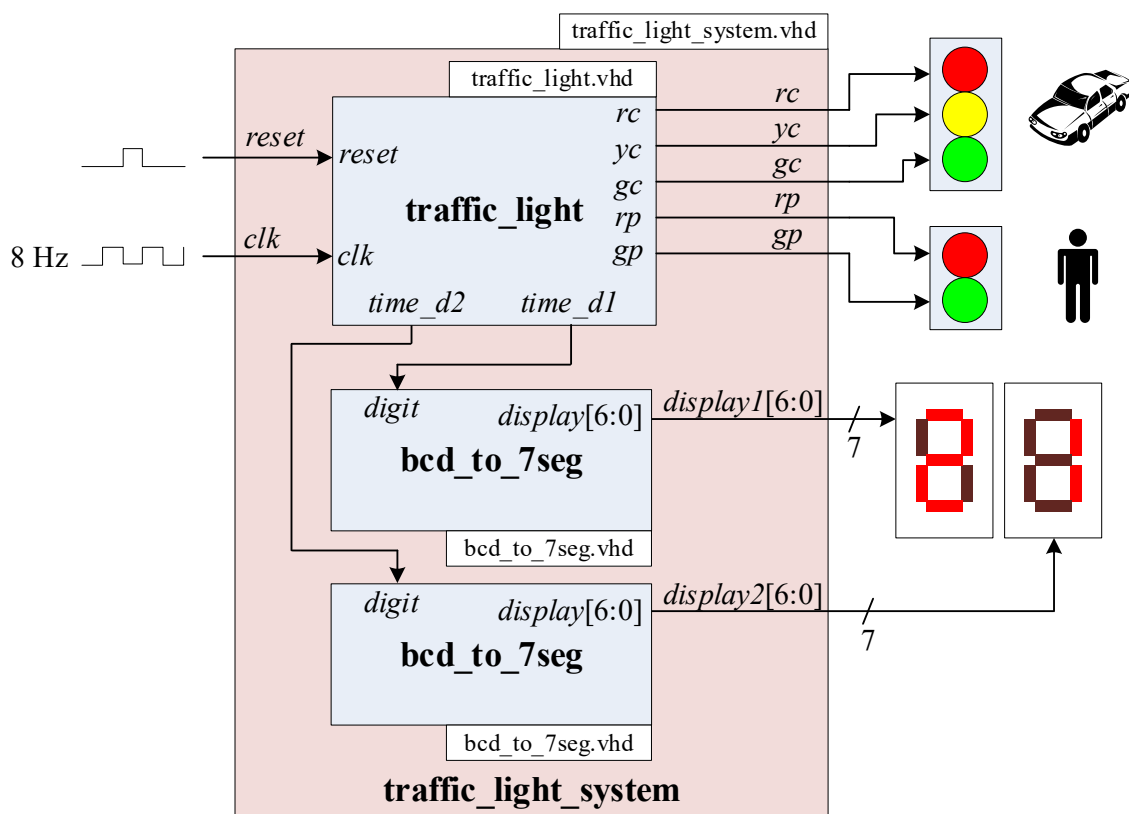
Слика 1 – Приказ симулатора семафора

2. Пројектовање хардвера у *VHDL*-у

На слици 2 је шематски приказан систем који потребно реализовати у *VHDL*-у.

Модул највише хијерархије треба назвати *traffic_light_system*. Он се састоји од три компоненте: контролера семафора, који је потребно сместити у компоненту

traffic_light, и два конвертора цифре из *BCD* кода у код за седмосегментни *LED* дисплеј, који је потребно сместити у компоненту *bcd_to_7seg*.



Слика 2 – Шематски приказ система за контролу рада семафора

2.1. Контролер семафора

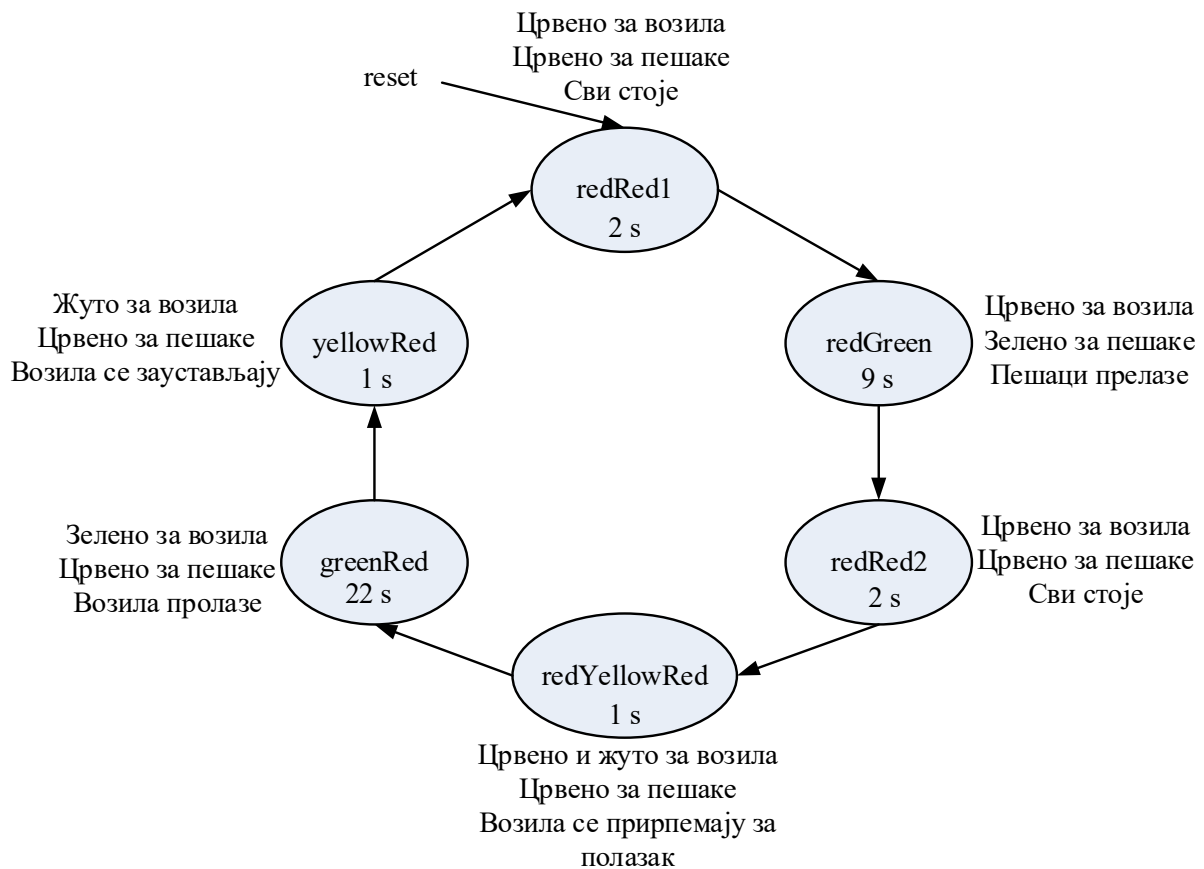
Контролер семафора треба реализовати као машину стања чији је дијаграм приказан на слици 3. Након сигнала ресета, активног у логичкој јединици, почетно стање семафора је црвено светло за пешаке и за возила. Након истека дефинисаног времена, контролер прелази у наредно стање. У сваком стању је потребно генерисати излазне сигнале за укључивање светала на семафорима (*rc* (*red for cars*) – црвено за возила, *yc* (*yellow for cars*) – жуто за возила, *gc* (*green for cars*) – зелено за возила, *rp* (*red for pedestrians*) – црвено за пешаке, *gp* (*green for pedestrians*) – зелено за пешаке). Излазни сигнали су активни у логичкој јединици.

У току стања *greenRed* (зелено за возила, црвено за пешаке) на излазима *time_d1* и *time_d2* треба приказати број секунди који је преостао до краја тог стања. Ови сигнали представљају две цифре броја секунди. Излази *time_d1* и *time_d2* могу бити типа *integer*, али и типа *std_logic_vector*. Поменути излази ће се превести у код за приказ на седмосегментним *LED* дисплејима. Број секунди се приказује једино у стању *greenRed*. У свим осталим стањима, сви сегменти на дисплејима треба да буду искључени.

Реализација приказа броја преосталих секунди може се извести на више начина. Студентима се оставља да одлуче који начин ће имплементирати. Овде се наводе две идеје. Први приступ је следећи. Бројач може да броји у природном бинарном коду, а да се реализује комбинациона логика којом се број у природном бинарном коду преводи у двоцифрен број у *BCD* коду. Други, занимљивији, а нимало тежи начин је да се бројање реализује преко два сигнала од којих је сваки посебна цифра и да се одмах

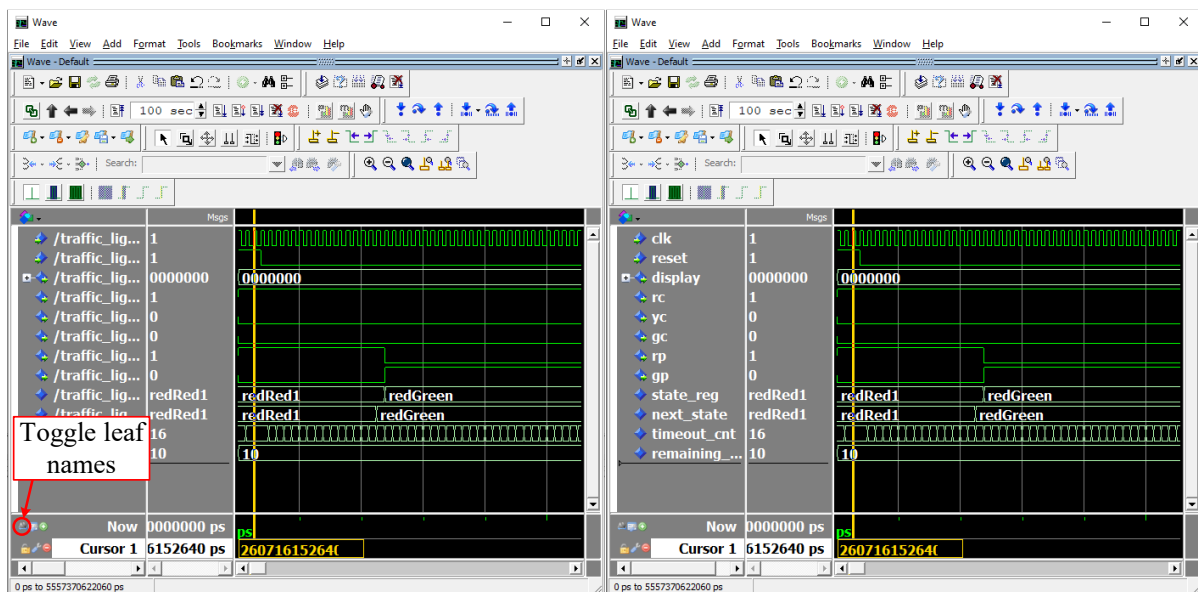
броји у BCD коду. У том случају се бројање за цифру веће тежине дозвољава само када цифра мање тежине треба да пређе са 0 на 9.

Сигнал такта је учестаности 8 Hz.



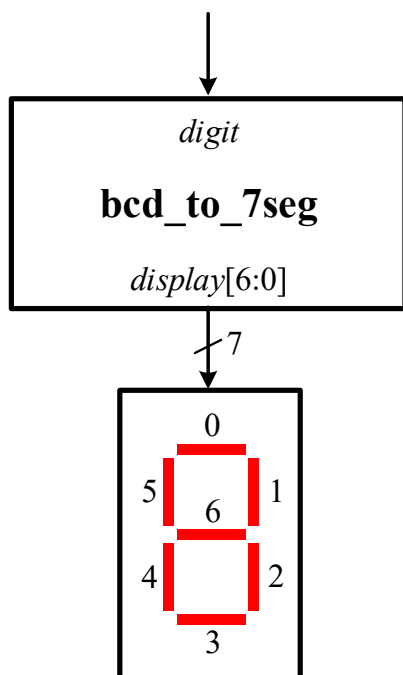
Слика 3 – Дијаграм стања контролера семафора

- Направити фајл *traffic_light.vhd* у коме треба описати у *VHDL*-у тражену машину стања. Приликом писања кода водити рачуна да називи портова одговарају ознакама са слике 2. За генерисање тражених временских интервала користити бројач реализован као помоћни сигнал типа *integer*.
 - Идеја: При свакој узлазној ивици сигнала такта проверавати да ли је бројач достигао одговарајућу вредност. С обзиром на то да машина стања треба да реализује и функционалност одбројавања секунди, препорука је да бројач броји уназад, а да му се почетна вредност дефинише при преласку у свако од наредних стања. У следеће стање се прелази ако је бројач достигао вредност 1.
- Креирати симулацију семафора којом треба испитати функционисање машине стања у симулатору *ModelSim-Intel FPGA*. Тест фајл назвати *traffic_light_tb.vhd*. На временском дијаграму јасно приказати све релевантне сигнале и прелазе између стања и дијаграм приказати у извештају. За прегледнији приказ сигнала на дијаграмима, кликнути на дугме *Toggle leaf names* као што је приказано на слици 4. Поновни клик на ово дугме враћа имена сигнала на пуне путање. Сигнале приказати као на слици 4 десно.



Слика 4 – Команда за прегледнији приказ сигнала у временским дијаграмима

2.2. Модул за конверзију цифре из *BCD* кода у код 7 сегмената



Слика 4 – Конверзија цифре у код за приказ на 7-сегментном LED дисплеју

хексадецималном запису (*radix: Hexadecimal*).

За приказ броја секунди који је преостао до истека црвеног светла за пешаке и зеленог за возила потребно је пројектовати компоненту за конверзију цифре из *BCD* кода у код за приказ на седмосегментном *LED* дисплеју. На слици 4 је приказана блок шема повезивања овог модула са седмосегментним дисплејом. Број поред сегмента одговара индексу у податку *display[6:0]*. *LE* диода са бројем *n* је укључена ако је *display[n]* на логичкој јединици.

- У фајлу *bcd_to_7seg.vhd* пројектовати компоненту *bcd_to_7seg* која омогућава приказ свих цифара од 0 до 9, а у случају да улазни податак није цифра од 0 до 9, све диоде на дисплеју треба да буду искључене. Улазни податак треба да буде истог типа као и излазни сигнали *time_d1* и *time_d2* из претходне тачке.

- Идеја: Користити *case* наредбу.

- У симулатору *ModelSim-Intel FPGA* креирати симулацију овог комбинационог модула. Тест фајл назвати *bcd_to_7seg_tb.vhd*. На временском дијаграму јасно приказати улазне и излазне сигнале и дијаграм приказати у извештају. Излазни вектор приказати у

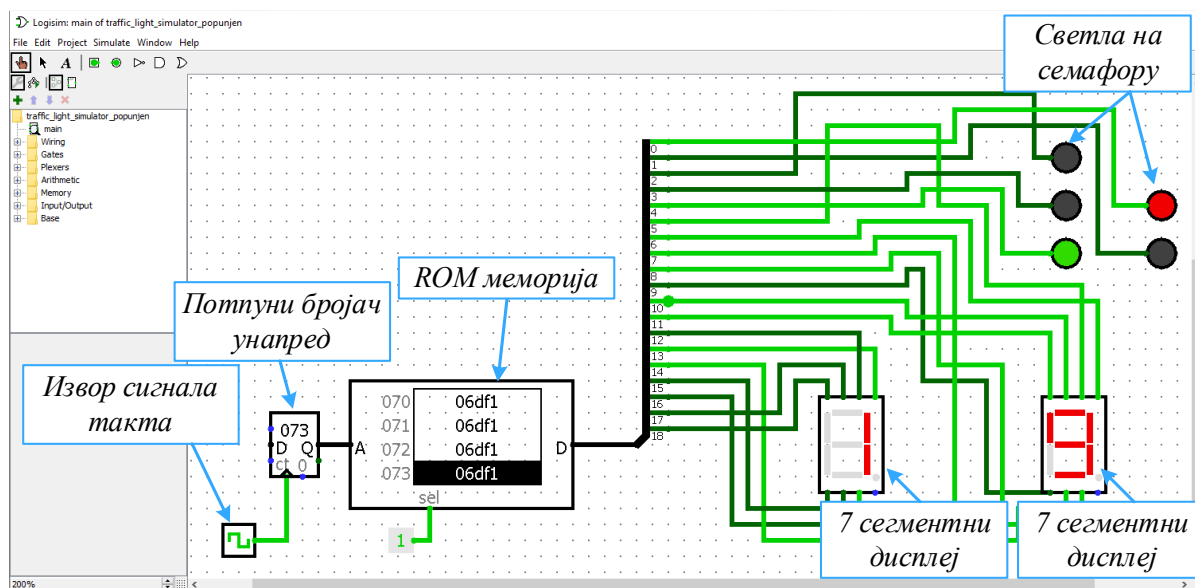
2.3. Повезивање целог система

Претходно пројектоване компоненте повезати у модул под именом *traffic_light_system* и сместити га у фајл *traffic_light_system.vhd*. У симулатору *ModelSim-Intel FPGA* симулирати овај систем коришћењем тест фајла који је приложен уз домаћи задатак *traffic_light_system_tb.vhd*. Симулацију урадити за 100 s (*ModelSim* препознаје секунде ако се запишу са „sec”).

Овај тест на сваку улазну ивицу сигнала такта све излазне сигнале модула *traffic_light_system* уписује у фајл на локацији *C:/romFile.txt*. Ови излазни сигнали ће бити учитани у визуелном симулатору контролера семафора реализованог у *Logisim* симулатору.

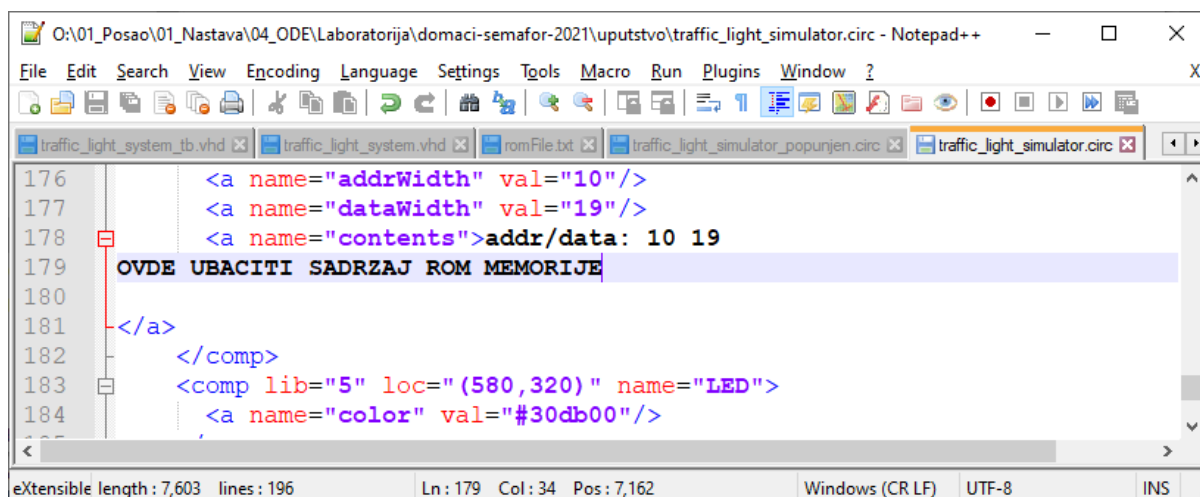
3. Приказ резултата симулације у *Logisim* симулатору

На слици 5 је приказан симулатор *Logisim* са отвореним пројектом за визуелизацију сигнала добијених у симулацији *VHDL* модула. Вредности ових сигнала су уписане у ROM меморију и при свакој улазној ивици сигнала такта се на излазу меморије појављују сигнали добијени у симулацији *VHDL* модула. Адреса меморије се мења коришћењем потпуног бројача унапред. Излаз меморије је 19-битни податак. Најнижих 5 бита овог податка представља вредности сигнала *rp*, *gp*, *rc*, *yc* и *gc*, док преосталих 14 бита представља вредности које је потребно довести на улазе *LED* дисплеја.



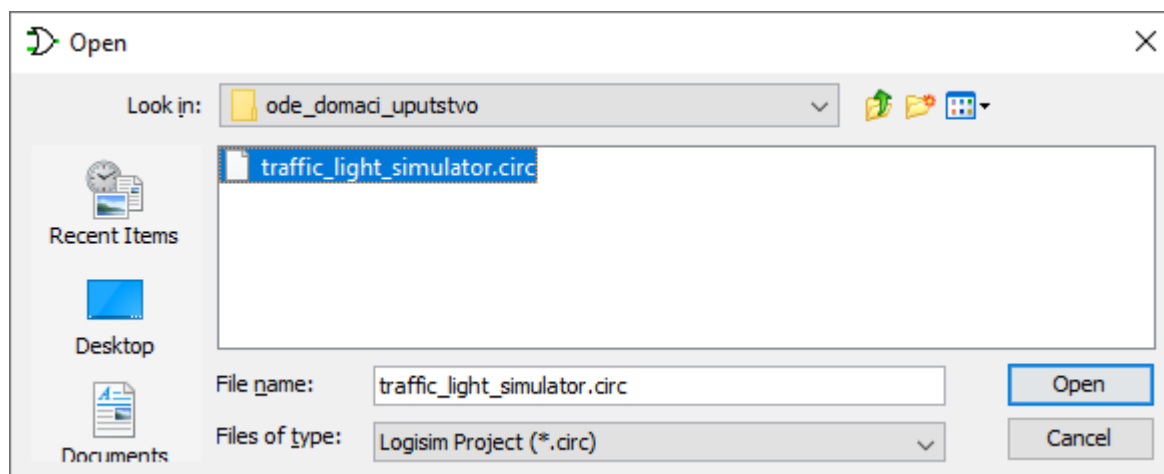
Слика 5 – Прозор симулатора за визуелизацију излазних сигнала система за контролу рада семафора

Фајл који је потребно учитати у *Logisim* симулатор је приложен уз овај домаћи задатак (*traffic_light_simulator.circ*). У фајл је пре учитавања потребно ископирати резултате симулације у *ModelSim*-у. Отворити фајл *traffic_light_simulator.circ* у неком алату за едитовање текстуалних фајлова (на пример *Notepad++*). Комплетан садржај фајла *C:/romFile.txt* ископирати у фајл *traffic_light_simulator.circ* почев од 179. линије (слика 6) – уместо текста „OVDE UBACITI SADRZAJ MEMORIJE“.



Слика 6 – Фајл у који је потребно ископирати резултате симулације

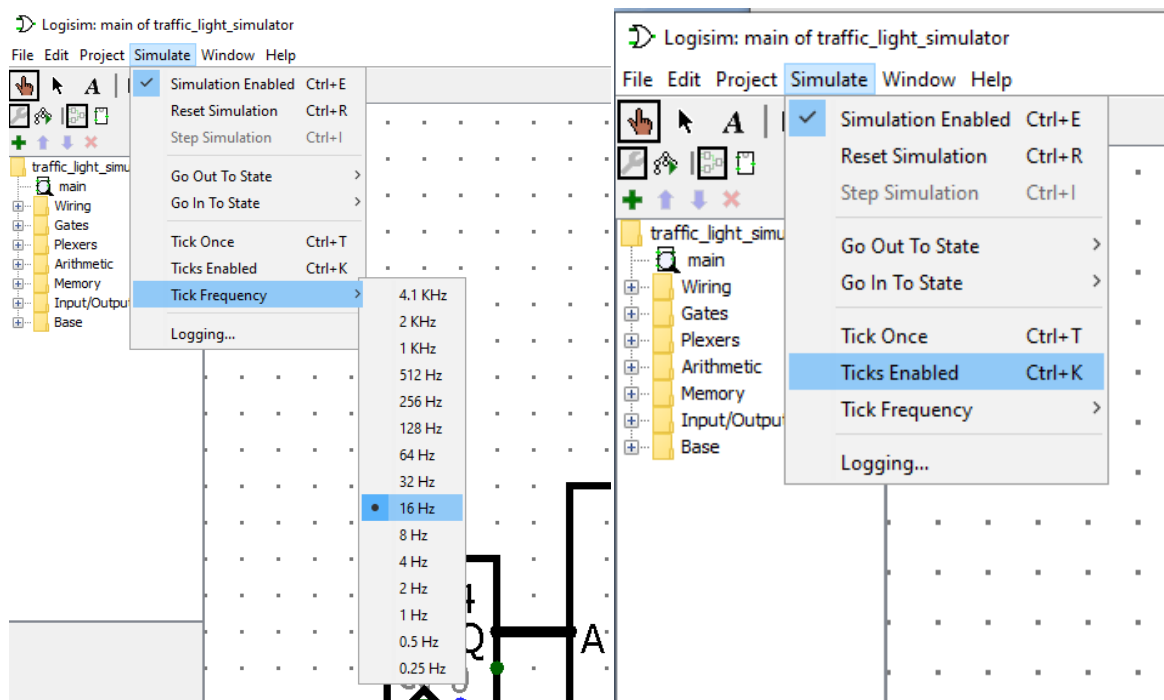
Отворити *Logisim* и кликом на *File→Open* отворити фајл *traffic_light_simulator.circ* (слика 7). Уколико садржај није добро копиран, алат неће успешно парсирати фајл и неће отворити описано коло.



Слика 7

Најпре подесити учестаност сигнала такта као на слици 8 – лево. Обратите пажњу да је овде *Tick Frequency* 16 Hz. Ово је учестаност сваке промене сигнала такта, па је зато она дупло већа о периоде сигнала такта. Након подешавања учестаности започети симулацију кликом на *Simulate→Ticks Enabled* (слика 8 – десно).

Видео у коме је приказан рад симулатора приложен је уз текст домаћег задатка.



Слика 8

4. Администрација

Домаћи задатак се ради у паровима и није обавезан. Успешно урађен домаћи задатак вреди 5 додатних поена који се сабирају са поенима освојеним на испиту и колоквијуму. Студенти се не пријављују за израду домаћег задатка.

Направљен је нови канал под називом „Домаћи задатак“ на коме студенти који немају пара могу да се јаве и тако пробају да нађу са ким ће радити. На овом каналу студенти могу постављати општа питања везана за домаћи задатак. Специфична питања која садрже детаље имплементације треба постављати предметном асистенту електронском поштом.

Попунити шаблон за извештај приложен уз текст овог задатка. Извештај и све .vhd фајлове запаковати у архиву и назвати је *ode_domaci_bbbb_gggg_bbbb_gggg.zip* (или .rar или .7z), где су *bbbb_gggg* бројеви индекса студената који шаљу решење. Послати предметном асистенту и наставнику електронском поштом.

Рок за слање решења је недеља **28. јун у 23.59.**

Одбране домаћих задатака ће бити организоване након овог рока. Детаљи о организацији одбране домаћих задатака биће објављени накнадно. Највероватније ће одбране бити у форми теста у термину испита у коме студент одлучи да полаже.