

Рачунарски ВЛСИ системи (13e114влси)

Прва предиспитна обавеза

20.11.2021.

Напомене:

- Време за израду решења је 120 минута.
- Решење се оцењује по принципу „ради или не ради” посматрано на нивоу сваке појединачне ставке.

Упутство:

- Виртуелну машину **W7_x64_VLSI_2021** која се налази на путањи **C:\TEMP** треба покренути помоћу алата **VMware Workstation 16 Player**.
- На виртуелној машини су доступни алати **Questa SIM** и **Visual Studio Code**.
- Након покретања виртуелне машине, неопходно је омогућити коришћење поменутих алата покретањем датотеке **RenewLicense.bat** која се налази на путањи **C:\questasim64_10.4c\win64**.
- На **Desktop** виртуелне машине треба прекопирати директоријум **Projekat** и отворити га десним кликом и одабиром опције **Open with Code**.
- У оквиру пројекта, на путањи **Projekat\src\simulation** налазе се:
 - празне датотеке које треба попунити **stavka_b.v** и **stavka_d.v**
 - датотеке са решењима **stavka_b_resenje.vp** и **stavka_d_resenje.vp**
 - датотеке са тестовима **stavka_a_test.vp** и **stavka_c_test.vp**
- У оквиру пројекта, на путањи **Projekat\src\simulation\modules** налазе се:
 - празне датотеке које треба попунити **stavka_a.v** и **stavka_c.v**
 - датотеке са решењима **stavka_a_resenje.vp** и **stavka_c_resenje.vp**
 - датотеке са тестовима **stavka_b_test.vp** и **stavka_d_test.vp**
- У оквиру пројекта, на путањи **Projekat\tooling** налази се:
 - датотека **makefile** у којој се уноси назив модула који се покреће
- У оквиру пројекта, на путањи **Projekat\tooling\config** налазе се:
 - датотека **list-src-files-simul.lst** у којој се наводе путање до датотека које се користе приликом покретања
 - датотека **waveform-define.do** у којој се наводе називи сигнала које **wave** треба да прикаже
 - датотека **run.tcl**
- Датотеке са решењима **stavka_a_resenje.vp** и **stavka_b_resenje.vp** се искључиво покрећу заједно.
- Датотеке са решењима **stavka_c_resenje.vp** и **stavka_d_resenje.vp** се искључиво покрећу заједно.
- Приликом покретања датотека са решењима неопходно је привремено закоментарисати прву линију датотеке **run.tcl** додавањем знака **#** на почетак реда.
- Тест за ставку се покреће заједно са одговарајућом ставком (нпр. **stavka_a_test.vp** и **stavka_a.v**).
- Приликом тестирања ставки **stavka_b.v** и **stavka_d.v** неопходно је привремено променити назив модула који се инстанцира на **stavka_b_test**, односно на **stavka_d_test**.
- На **Rad (L:)** предају се искључиво датотеке **stavka_a.v**, **stavka_b.v**, **stavka_c.v** и **stavka_d.v**.

Задатак:

(a) [7 поена] Написати комбинациони модул **stavka_a** у оквиру засебне датотеке **stavka_a.v** који има следеће портове:

- једнобитни контролни улазни сигнал **control**
- осмобитни улазни податак **repr**
- једнобитни излазни податак **sign**
- осмобитни излазни податак **value**

Модул треба да за задату представу броја одреди вредност тог броја. Контролни сигнал **control** даје информацију о томе у ком комплементу је број представљен. Број је представљен у првом комплементу када сигнал **control** има вредност нула, односно у другом комплементу када сигнал **control** има вредност један. Улазни податак **repr** садржи представу броја. Излазни податак **sign** треба да садржи информацију о знаку броја. Ако је број позитиван **sign** треба да има вредност нула, а ако је број негативан **sign** треба да има вредност један. Излазни податак **value** треба да садржи вредност броја.

(b) [3 поена] Написати модул **stavka_b** за тестирање (*testbench*) у оквиру засебне датотеке **stavka_b.v** који инстанцира комбинациони модул **stavka_a** описан у претходној ставци (*DUT*). *Testbench* треба да побуди *DUT* са свим могућим улазним вредностима. *Testbench* треба да прати вредности свих улазних и излазних сигнала за *DUT* и у тренутку њихове промене врши испис симулационог тренутка и вредности свих улазних и излазних сигнала за *DUT*.

(c) [7 поена] Написати секвенцијални модул **stavka_c** у оквиру засебне датотеке **stavka_c.v** који има следеће портове:

- асинхрони ресет активан за вредност нула **rst_n**
- сигнал такта **clk**
- једнобитни контролни сигнал **inc**
- једнобитни контролни сигнал **dec**
- дванаестобитни излазни податак **data_out**.

Модул представља регистар који броји по Фибоначијевој секвенци (0, 1, 1, 2, 3, 5, 8, ...). Излазни податак након асинхроног ресета има вредност 0. Излазни податак мења вредност на улазну ивицу сигнала такта. Уколико контролни сигнал **inc** има вредност један, вредност излазног податка у наредној периоди сигнала такта се увећава на следећи елемент Фибоначијевог низа. Уколико контролни сигнал **dec** има вредност један, вредност излазног податка у наредној периоди сигнала такта се смањује на претходни елемент Фибоначијевог низа. Операција **inc** има већи приоритет од операције **dec**. Највећа вредност коју излазни податак може да има је 1597 (вредност излазног податка се не увећава преко 1597 без обзира на вредност контролног сигнала **inc**). Најмања вредност коју излазни податак може да има је 0 (вредност излазног податка се не смањује испод 0 без обзира на вредност контролног сигнала **dec**).

(d) [3 поена] Написати модул **stavka_d** за тестирање (*testbench*) у оквиру засебне датотеке **stavka_d.v** који инстанцира секвенцијални модул **stavka_c** описан у претходној ставци (*DUT*). *Testbench* треба да побуди *DUT* са стотину псеудослучајних улазних вредности. *Testbench* треба да испишује симулациони тренутак и вредности свих улазних и излазних сигнала за *DUT* на сваку улазну ивицу сигнала такта. За испис користити системски таск **\$strobe**. Овај системски таск прима исте аргументе као системски таск **\$display**.