

Рачунарски ВЛСИ системи (13e114влси)

Пример друге предиспитне обавезе

14.12.2021.

Напомене:

- Време за израду решења је 180 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, Quartus II** 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:
 - датотеке са тестовима **stavka_a_test.vp, stavka_b_test.vp, ..., stavka_f_test.vp**
- У оквиру пројекта, на путањи **Projekat\src\simulation\modules** налазе се:
 - празне датотеке које треба попунити **stavka_a.v, stavka_b.v, ..., stavka_f.v**
 - датотеке са решењима **stavka_a_resenje.vp, stavka_b_resenje.vp, ..., stavka_f_resenje.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 симулације
 - датотека **list-src-files-synth.lst** у којој се наводе путање до датотека које се користе приликом покретања синтезе
 - датотека **waveform-define.do** у којој се наводе називи сигнала које **wave** треба да прикаже
 - датотека **run.tcl**
- Приликом покретања тестова неопходно је привремено закоментарисати прву линију датотеке **run.tcl** додавањем знака **#** на почетак реда.
- Тест за ставку се покреће заједно са решењем ставке и са одговарајућом ставком (нпр. **stavka_a_test.vp, stavka_a_resenje.vp** и **stavka_a.v**).
- Приликом покретања синтезе неопходно је направити привремену копију модула за који се врши синтеза на путању **Projekat\src\synthesis\modules**.
- Анализа и синтеза за модул се покрећу наредбом **make synth_map**.
- На **Rad (L:)** предају се искључиво датотеке **stavka_a.v, stavka_b.v, stavka_c.v, stavka_d.v, stavka_e.v** и **stavka_f.v**.

Задатак:

(a) [5 поена] Написати модул **stavka_a** у оквиру засебне датотеке **stavka_a.v** који има следеће портове:

- асинхрони ресет активан за вредност нула **rst_n**
- сигнал такта **clk**
- једнобитни улазни податак **in**
- једнобитни излазни податак **out**

Модул треба да врши детекцију узлазне ивице на улазном сигналу **in**. Модул треба реализовати као мрежу Муровог типа (излаз зависи искључиво од тренутног интерног стања). Модул на излазу **out** генерише импулс, односно активну вредност сигнала у трајању од тачно једне периоде сигнала такта, у најранијем могућем тренутку (имајући у виду да се ради о мрежи Муровог типа) након детекције узлазне ивице на улазном сигналу. Излазни сигнал у осталим случајевима има неактивну вредност.

(b) [5 поена] Написати модул **stavka_b** у оквиру засебне датотеке **stavka_b.v** који има следеће параметре и портове:

- параметар **NUM**
- асинхрони ресет активан за вредност нула **rst_n**
- сигнал такта **clk**
- једнобитни улазни податак **in**
- једнобитни излазни податак **out**

Модул представља проширење модула из ставке (a) али за потребе оцењивања мора бити реализован као нови модул у засебној датотеци.

Модул треба реализовати као мрежу Муровог типа (излаз зависи искључиво од тренутног интерног стања). Модул поседује интерни бројач који се инкрементира у свакој периоди сигнала такта када улазни сигнал **in** има активну вредност. Вредност бројача се ресетује приликом детекције узлазне ивице на улазном сигналу **in** и приликом достизања вредности $6 * NUM$. Модул на излазу **out** генерише импулс, односно активну вредност сигнала у трајању од тачно једне периоде сигнала такта, приликом детекције силазне ивице на улазном сигналу **in** уколико је текућа вредност бројача већа или једнака вредности $3 * NUM$.

(c) [5 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 **select**
- једнобитни контролни сигнал **add**
- четворобитни улазни податак **data_in**
- четворобитни излазни податак **data_out**

Модул представља бафер са капацитетом за 2 једноцифрена броја. Податак са улаза **data_in** сабира се по модулу 10 са вредношћу која се налази у баферу на тренутно одабраној позицији када контролни сигнал **add** има активну вредност. Операција иницирана контролним сигналом **add** траје максимално једну периоду сигнала такта. Позиција у баферу на коју се додаје вредност одабира се

контролним сигналом **select**. Уколико контролни сигнал **select** има активну вредност одабрана је позиција 1 бафера, односно ако има неактивну вредност одабрана је позиција 0 бафера. Модул на излаз **data_out** поставља једноцифрени број који се налази на тренутно одабраној позицији бафера као четворобитни податак.

Имајући у виду да је предвиђено да се као контролни сигнали користе дугмићи и прекидачи у модулу треба инстанцирати и на одговарајући начин повезати детекторе улазних ивица и остале помоћне модуле.

(d) [5 поена] Написати секвенцијални модул **stavka_d** у оквиру засебне датотеке **stavka_d.v** који има следеће портове:

- асинхрони ресет активан за вредност нула **rst_n**
- сигнал такта **clk**
- једнобитни контролни сигнал **select**
- једнобитни контролни сигнал **add**
- једнобитни контролни сигнал **next**
- четворобитни улазни податак **data_in**
- четворобитни излазни податак **data_out**

Модул представља проширење модула из ставке **(c)** али за потребе оцењивања мора бити реализован као нови модул у засебној датотеци.

Модул функционише идентично као модул из ставке **(c)** све док контролни сигнал **next** не добије активну вредност, када модул мења понашање. У случају активне вредности контролног сигнала **next**, за једноцифрене бројеве који се тренутно налазе у баферу израчунава се највећи заједнички делилац. Такође, модул на излаз поставља једноцифрени број који се добија дељењем једноцифреног броја са тренутно одабране позиције бафера са израчунатим највећим заједничким делиоцем. Позиција у баферу одабира се контролним сигналом **select**, на исти начин као што је описано у ставци **(c)**. Контролни сигнал **add** нема улогу у овом режиму рада. Поновна активна вредност контролног сигнала **next** чини да се модул враћа у режим рада описан у ставци **(c)**.

Имајући у виду да је предвиђено да се као контролни сигнали користе дугмићи и прекидачи у модулу треба инстанцирати и на одговарајући начин повезати детекторе улазних ивица и остале помоћне модуле.

(e) [5 поена] Написати секвенцијални модул **stavka_e** у оквиру засебне датотеке **stavka_e.v** који има следеће параметре и портове:

- параметар **NUM**
- асинхрони ресет активан за вредност нула **rst_n**
- сигнал такта **clk**
- једнобитни контролни сигнал **select**
- једнобитни контролни сигнал **add**
- једнобитни контролни сигнал **next**
- четворобитни улазни податак **data_in**
- четворобитни излазни податак **data_out**

Модул представља проширење модула из ставке **(d)** али за потребе оцењивања мора бити реализован као нови модул у засебној датотеци.

Модул функционише идентично као модул из ставке (d) осим у погледу поновне активне вредности контролног сигнала **next**. Модул се не враћа у режим рада у ставци (c), већ започиње декрементирање од израчунате вредности највећег заједничког делиоца (која је смештена у помоћни регистар) на сваких **NUM** периода сигнала такта све док помоћни регистар не добије вредност нула. Када помоћни регистар добије вредност нула модул се поново понаша као модул из ставке (c). У режим рада описан у ставци (c) могуће је прећи и пре завршетка декрементирања у случају активне вредности контролног сигнала **next** у тренутку када је вредност у помоћном регистру делилац једноцифрених бројева из бафера. Модул на излаз **data_out** поставља сигнал чија је вредност једнака вредности у помоћном регистру.

Имајући у виду да је предвиђено да се као контролни сигнали користе дугмићи и прекидачи у модулу треба инстанцирати и на одговарајући начин повезати детекторе узлазне ивице и остале помоћне модуле.

(f) [5 поена] Написати секвенцијални модул **stavka_f** у оквиру засебне датотеке **stavka_f.v** који има следеће параметре и портове:

- параметар **NUM**
- асинхрони ресет активан за вредност нула **rst_n**
- сигнал такта **clk**
- једнобитни контролни сигнал **select**
- једнобитни контролни сигнал **add**
- једнобитни контролни сигнал **next**
- четворобитни улазни податак **data_in**
- седмобитни излазни податак **data_out**

Модул представља проширење модула из ставке (e) али за потребе оцењивања мора бити реализован као нови модул у засебној датотеци.

Модул функционише идентично као модул из ставке (e) осим у погледу контролног сигнала **add** и формата излаза. Податак са улаза **data_in** сабира се по модулу 10 са вредношћу која се налази у баферу на тренутно одабраној позицији, када контролни сигнал **add** има активну вредност у трајању од бар $3 * \text{NUM}$ периода сигнала такта. У случају трајања активне вредности на контролном сигналу **add** дужем од $6 * \text{NUM}$ периода сигнала такта, мерење трајања активне вредности на контролном сигналу **add** почети испочетка (од вредности 0). Уместо бинарне представе вредности помоћног регистра излаз модула је седмобитни податак који представља представу децималне цифре за приказ на седмосегментном екрану.

Имајући у виду да је предвиђено да се као контролни сигнали користе дугмићи и прекидачи у модулу треба инстанцирати и на одговарајући начин повезати детекторе узлазне ивице и остале помоћне модуле.