

Рачунарски ВЛСИ системи (13e114влси)

Пример друге предиспитне обавезе

Напомене:

Решење се оцењује по принципу “ради или не ради” посматрано на нивоу сваке појединачне ставке.

Задатак:

Користећи алате *QuestaSim* и *Quartus II* потребно је испунити ставке које се налазе у наставку. Приликом израде решења на располагању је документација *Verilog* језика. Иницијално стање свих флип-флопова након асинхроног ресета јесте неактивно осим уколико поставком задатка на одређеним местима није другачије наглашено.

(a) [5 поена] Написати модул имена **stavka_a** у оквиру засебне датотеке **stavka_a.v** који има следеће портове: сигнал такта (улаз) **clk**, асинхрони ресет активан у вредности нула (улаз) **rst_n**, једнобитни сигнал (улаз) **in** и једнобитни сигнал (излаз) **out**. Модул врши детекцију узлазне ивице на улазном сигналу. Модул реализовати као мрежу Муровог типа (излаз зависи искључиво од тренутног интерног стања). Модул на излазу 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.

(б) [5 поена] Написати модул имена **stavka_b** у оквиру засебне датотеке **stavka_b.v** који има следеће портове: сигнал такта (улаз) **clk**, асинхрони ресет активан у вредности нула (улаз) **rst_n**, једнобитни сигнал (улаз) **in** и једнобитни сигнал (излаз) **out**. Модул представља *debouncer* који елиминира подрхтавање сигнала. Модул реализовати као мрежу Муровог типа (излаз зависи искључиво од тренутног интерног стања). Модул на излазу генерише сигнал чија је вредност једнака последњој стабилној вредности улазног сигнала. Сматрати да је вредност улазног сигнала стабилна уколико се није мењала у претходних 256 периода сигнала такта.

(ц) [5 поена] Написати модул имена **stavka_c** у оквиру засебне датотеке **stavka_c.v** који има следеће портове: сигнал такта (улаз) **clk**, асинхрони ресет активан у вредности нула (улаз) **rst_n**, четворобитни податак (улаз) **data_in**, једнобитни контролни сигнал (улаз) **inc**, једнобитни контролни сигнал (улаз) **ld** и четворобитни податак (излаз) **data_out**. Модул представља кружни бафер са капацитетом за 3 четворобитна податка. Податак са улаза се уписује, када контролни сигнал **ld** има активну вредност, у кружни бафер на тренутну позицију за упис. Уписом податка у кружни бафер позиција за упис помера се на наредну позицију у кружном баферу. Кружни бафер не води рачуна о својој попуњености и може прегазити постојеће податке у случају уписа нових. Позицију за упис могуће је померити на наредну позицију и без уписа новог податка, односно без гажења вредности у кружном баферу на тренутној позицији за упис, помоћу активне вредности контролног сигнала **inc**. Операције инициране контролним сигнаlima **ld** и **inc** трају максимално једну периоду сигнала такта. Модул на излаз поставља четворобитни податак чија вредност одговара податку из кружног бафера са тренутне позиције за упис (податак који би био прегажен у

случају уписа). Уколико се позиција за упис промени услед контролног сигнала **inc** потенцијално се мења и излаз модула. Модул треба инстанцирати у оквиру *Top-Level Entity* који описује развојну плочицу са *FPGA* чипом и извршити повезивање на један од следећа два начина у зависности од коришћене плочице:

```
// DE0_CV_TOP -> Cyclone V
stavka_c inst_c (
    .clk(CLOCK_50),
    .rst_n(SW[9]),
    .data_in(SW[3:0]),
    .ld(SW[4]),
    .inc(~KEY[0]),
    .data_out(LED[3:0])
);

// DE0_TOP -> Cyclone III
stavka_c inst_c (
    .clk(CLOCK_50),
    .rst_n(SW[9]),
    .data_in(SW[3:0]),
    .ld(SW[4]),
    .inc(~BUTTON[0]),
    .data_out(LEDG[3:0])
);
```

Имајући у виду да се као контролни сигнали користе дугмићи и прекидачи у модулу **stavka_c** треба инстанцирати и на одговарајући начин повезати *debouncer* и детекторе узлазне ивице.

(д) [5 поена] Написати модул имена **stavka_d** у оквиру засебне датотеке **stavka_d.v** који има следеће портове: сигнал такта (улаз) **clk**, асинхрони ресет активан у вредности нула (улаз) **rst_n**, четворобитни податак (улаз) **data_in**, једнобитни контролни сигнал (улаз) **inc**, једнобитни контролни сигнал (улаз) **ld** и четворобитни податак (излаз) **data_out**. Модул представља проширење модула из ставке (ц) али за потребе оцењивања мора бити реализован као нови модул у засебној датотеци. Модул функционише идентично као модул из ставке (ц) све док макар један податак у кружном баферу има вредност нула. Када се у кружном баферу нађу сва 3 податка са вредностима различитим од нуле контролни сигнали **ld** и **inc** добијају другачије функције и модул генерише излаз на другачији начин. У зависности од вредности контролног сигнала **ld** израчунава се резултат одговарајуће битске операције над сва 3 податка из кружног бафера и дати резултат се чува у помоћном регистру. Уколико контролни сигнал **ld** има активну односно неактивну вредност у помоћном регистру се чува резултат битске операције OR односно AND, респективно. Модул на излаз поставља сигнал чија је вредност једнака вредности у помоћном регистру. Активна вредност контролног сигнала **inc** чини да се модул поново понаша као модул из ставке (ц) (том приликом сва 3 податка у кружном баферу се постављају на вредност нула).

(е) [5 поена] Написати модул имена **stavka_e** у оквиру засебне датотеке **stavka_e.v** који има параметар **NUM** и следеће портове: сигнал такта (улаз) **clk**, асинхрони ресет активан у вредности нула (улаз) **rst_n**, четворобитни податак (улаз) **data_in**, једнобитни контролни сигнал (улаз) **inc**, једнобитни контролни сигнал (улаз) **ld** и четворобитни податак (излаз) **data_out**. Модул представља проширење модула из ставке (д) али за потребе оцењивања мора бити реализован као нови модул у засебној датотеци. Модул функционише идентично као модул из ставке (д) осим у погледу контролног сигнала **inc** након тренутка у којем сва 3 податка у кружном баферу постану различита од нуле. Активна вредност контролног сигнала **inc**, након тренутка у којем сва 3 податка у кружном баферу постану различита од нуле, започиње декрементирање вредности помоћног регистра 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 (ц) (овом приликом сва 3 податка у кружном баферу се постављају на вредност нула). Приликом рада на развојној плочици са *FPGA* чипом, параметар **NUM** одабрати тако да се декрементирање врши након сваке протекле секунде. У случају коришћења алата *QuestaSim* прилагодити параметар **NUM** тако да одговара процесу симулације.

(ф) [5 поена] Написати модул имена **stavka_f** у оквиру засебне датотеке **stavka_f.v** који има параметар **NUM** и следеће портове: сигнал такта (улаз) **clk**, асинхрони ресет активан у вредности нула (улаз) **rst_n**, четворобитни податак (улаз) **data_in**, четворобитни податак (улаз) **data_in**, једнобитни контролни сигнал (улаз) **inc**, једнобитни контролни сигнал (улаз) **ld** и седмобитни податак (излаз) **data_out**. Модул представља проширење модула из ставке (е) али за потребе оцењивања мора бити реализован као нови модул у засебној датотеци. Модул функционише идентично као модул из ставке (е) осим у погледу формата излаза. Уместо бинарне представе вредности помоћног регистра излаз модула је седмобитни податак који представља представу хексадецималне цифре за приказ на седмосегметном екрану. Модул треба инстанцирати у оквиру *Top-Level Entity* који описује развојну плочицу са *FPGA* чипом и извршити повезивање на један од следећа два начина у зависности од коришћене плочице:

```
// DE0_CV_TOP -> Cyclone V
stavka_f inst_f (
    .clk(CLOCK_50),
    .rst_n(SW[9]),
    .data_in(SW[3:0]),
    .ld(SW[4]),
    .inc(~KEY[0]),
    .data_out(HEX0)
);
```

```
// DE0_TOP -> Cyclone III
stavka_f inst_f (
    .clk(CLOCK_50),
    .rst_n(SW[9]),
    .data_in(SW[3:0]),
    .ld(SW[4]),
    .inc(~BUTTON[0]),
    .data_out(HEX0_D)
);
```