

# Рачунарски ВЛСИ системи (13e114влси)

## Пример прве предиспитне обавезе

### Напомене:

Приликом израде решења на располагању је документација *Verilog* језика, алати *Questa SIM* и *Visual Studio Code*, криптована решења ставки и криптовани тестови ставки.

На *Rad (L:)* предају се искључиво *Verilog* датотеке **stavka\_a.v**, **stavka\_b.v**, **stavka\_c.v** и **stavka\_d.v**.

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

### Комбинациони модул:

(а) [7 поена] Написати комбинациони модул **stavka\_a** у оквиру засебне датотеке **stavka\_a.v** који има следеће портове:

- седмобитни улазни податак **data\_in**
- једнобитни контролни улазни сигнал **control**
- осмобитни излазни податак **data\_out**

Модул треба да утврди да ли битска репрезентација улазног седмобитног податка садржи више нула или јединица и да на излаз проследи дати седмобитни податак заједно са утврђеном једнобитном информацијом уметнутом између трећег и четвртог бита седмобитног податка. Уколико контролни сигнал има вредност нула, једнобитна информација има вредност један у случају да битска репрезентација улазног седмобитног податак садржи више нула, док у супротном има вредност нула. Уколико контролни сигнал има вредност један, једнобитна информација има вредност један у случају да битска репрезентација улазног седмобитног податак садржи више јединица, док у супротном има вредност нула.

(б) [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* исписује симулациони тренутак и вредности свих улазних и излазних сигнала за *DUT* при свакој промени излазног сигнала.

### Секвенцијални модул

(с) [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**
- четворобитни улазни податак **data\_in**
- тробитни контролни улазни сигнал **control**
- четворобитни излазни податак **data\_out**.

Излазни податак након асинхроног ресета има вредност нула. Излазни податак мења вредност на улазну ивицу сигнала такта. Битови контролног сигнала имају следеће значење: нулти бит представља *enable* бит, први бит представља *double* бит (одређује да ли је потребно дуплирати четворобитни податак са улаза пре његовог коришћења у операцији) и други бит представља *operation* бит (дефинише коју операцију треба извршити). Уколико *enable* бит има вредност нула, вредност на излазу у наредној периоди сигнала такта остаје непромењена независно од вредности осталих улазних сигнала. Вредност операнда операције јесте (1) вредност улазног четворобитног податка уколико *double* бит има вредност нула или (2) дуплирана вредност улазног четворобитног податка уколико *double* бит има вредност један. Вредност на излазу модула у наредној периоди сигнала такта јесте (1) вредност операнда уколико *operation* бит има вредност нула или (2) вредност на излазу у тренутној периоди сигнала такта увећана за један уколико *operation* бит има вредност један.

**(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 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*.