

# Peta laboratorijska vježba

## Osnovi računarske tehnike

### UVOD

Ova laboratorijska vježba za cilj ima upoznavanje studenata mašinama sa konačnim brojem stanja, načinom njihovog funkcionalnog opisa u *Verilog* jeziku za opis hardvera i njihovom simulacijom u programskom paketu *Quartus*.

### PRIPREMA

**Zadatak 1.** Projektovati mašinu sa konačnim brojem stanja za sekvencijalnu mrežu kojom se realizuje funkcija uključenja/isključenja nekog izlaznog uređaja (npr. dioda koja emituje svjetlost – LED dioda). Mašinu predstaviti odgovarajućim dijagramom stanja.

**Zadatak 2.** Projektovati mašinu sa konačnim brojem stanja za sekvencijalnu mežu kojom se realizuje funkcija obostranog pomjeračkog registra.

**Zadatak 3.** Projektovati mašinu sa konačnim brojem stanja za sekvencijalnu mrežu kojom se realizuje funkcija obostranog brojača modula  $M=5$ .

Napomena: *Urađenu pripremu, na listu papira, donijeti sa sobom na termin laboratorijske vježbe. Ukoliko ne donesete pripremu, nećete moći prisustvovati laboratoriskoj vježbi.*

# RAD U LABORATORIJI

## Zadatak 1:

Koristeći tekstualni editor *Notepad/Notepad++* otvoriti tekstualni fajl **task\_1.v** koji se nalazi unutar vašeg repozitorijuma na putanji “lab\_5/hdl/”. Unijeti opis logičkog kola projektovanog u zadatku 1 (priprema). Pokrenuti simulaciju, prikazati ulaze i izlaze datog logičkog kola i prokomentarisati dobijene rezultate.

## Zadatak 2:

Koristeći tekstualni editor *Notepad/Notepad++* otvoriti tekstualni fajl **task\_2.v** koji se nalazi unutar vašeg repozitorijuma na putanji “lab\_5/hdl/”. Unijeti opis logičkog kola projektovanog u zadatku 2 (priprema). Pokrenuti simulaciju, prikazati ulaze i izlaze datog logičkog kola i prokomentarisati dobijene rezultate.

## Zadatak 3:

Koristeći tekstualni editor *Notepad/Notepad++* otvoriti tekstualni fajl **task\_3.v** koji se nalazi unutar vašeg repozitorijuma na putanji “lab\_5/hdl/”. Unijeti opis logičkog kola projektovanog u zadatku 3 (priprema). Pokrenuti simulaciju, prikazati ulaze i izlaze datog logičkog kola i prokomentarisati dobijene rezultate.

# DODATAK

## Pokretanje simulacije

Simulacija se pokreće na sljedeći način.

Nakon što ste uspješno preuzeli materijale za izradu laboratorijske vježbe (tako što ste uradili *clone* vašeg repozitorijuma ili *pull* unutar vašeg već *clone*-iranog repozitorijuma) pozicionirajte se unutar **sim** direktorijuma na sljedeći način:

**Napomena:** *Prije nego što krenete sa simulacijama, potrebno je izvršiti skriptu “env\_config.bat” koja se nalazi unutar vašeg repozitorijuma. Preporučujem da ovaj korak uradite na samom početku rada.*

Iz vašeg radnog direktorijuma **ort-<vaš\_broj\_indeksa>** (npr. ako je vaš broj indeksa 1234/56, naziv vašeg radnog direktorijuma je **ort-123456**) u **command prompt**-u izvršite niz sljedećih naredbi:

```
env_config.bat
cd lab_5
cd sim
dir
```

Nakon izlistavanja sadržaja **sim** direktorijuma, trebalo bi da imate slj. ispis (Napomena: Prikazani ispis je djelimičan, moguće je da će se još fajlova nalaziti unutar **sim** direktorijuma):

10/29/2023 09:35 AM	<DIR>	.
10/29/2023 09:35 AM	<DIR>	..
10/29/2023 09:35 AM		48 clean.sh
10/29/2023 09:35 AM	<DIR>	lists
10/29/2023 09:35 AM		286 run_task_1.do
10/29/2023 09:35 AM		287 run_task_2.do
10/29/2023 09:35 AM		286 run_task_3.do
10/29/2023 09:35 AM	<DIR>	waves
	6 File(s)	1,478 bytes
	4 Dir(s)	405,903,982,592 bytes free

Nakon što ste se uspješno pozicionirali unutar **sim** direktorijuma, simulaciju ćete pokrenuti tako što ćete izvršiti jednu od sljedećih komandi:

<ul style="list-style-type: none"> <li>Za logičko kolo iz zadatka 1 <code>vsim -do run_task_1.do</code></li> </ul>	<ul style="list-style-type: none"> <li>Za logičko kolo iz zadatka 2 <code>vsim -do run_task_2.do</code></li> </ul>
<ul style="list-style-type: none"> <li>Za logičko kolo iz zadatka 3 <code>vsim -do run_task_3.do</code></li> </ul>	

## Verilog operatori

U tabeli 1. su dati *bit-wise* operatori u *Verilog-u*

Operator	Opis
~	Negacija ("NOT", "NE")
&	"AND" – Bitski "I" (*)
	"OR" – Bitsko "ILI" (+)
^	"XOR" - Ekskluzivno "ILI"
{}	Konkatenacija – Npr. <pre>wire a, b;           // 1-bit wire wire [1 : 0] res;    // 2-bit wire to store a and b  // res[1] follows a, // res[0] follows b assign res = {a, b};</pre>

## Verilog naredbe

IF-ELSE-IF	CASE
<pre>if (condition_1) begin     statement 1;     ...     statement k; end else if (condition_2) begin     statement 1;     ...     statement k; end else if (condition_3) begin     ...     ... end else begin     statement 1;     ...     statement k; end</pre>	<pre>case (expression)     &lt;case_1&gt; : begin         statement 1;         ...         statement k;     end     &lt;case_2&gt; : begin         statement 1;         ...         statement k;     end     ...     default : begin         statement 1;         ...         statement k;     end end endcase</pre>