



Univerza v Ljubljani

Fakulteta  
za računalništvo  
in informatiko



# Metode logičnega snovanja

## Sekvenčna vezja v jeziku VHDL (2. del)

Miha Moškon



# Pomikalni register (Shift register)

Vsako urino periodo biti v registru zamenjajo svojo pozicijo na naslednji način:

- pomik v levo:
  - $q(0) = q(n)$
  - $q(1) = q(0)$
  - ...
  - $q(n) = q(n-1)$
- pomik v desno:
  - $q(0) = q(1)$
  - $q(1) = q(2)$
  - ...
  - $q(n) = q(0)$

# Pomikalni register (Shift register)

Opis z vektorskimi tipi lahko izkorišča operacijo konkatencije:

- pomik v levo:
  - $q = q(n-1 \text{ downto } 0) \ \& \ q(n)$
- pomik v desno:
  - $q = q(0) \ \& \ q(n \text{ downto } 1)$

# Števec

Opis štetja navzgor z logičnimi enačbami:

- $q(0) = q(0) \text{ xor } 1 = \text{not } q(0)$
- $q(1) = q(1) \text{ xor } q(0)$
- $q(2) = q(2) \text{ xor } (q(1) \text{ and } q(0))$
- ...
- $q(n) = q(n) \text{ xor } (q(n-1) \text{ and } \dots \text{ and } q(0))$

Opis štetja navzdol na podoben način.

# Naloga

- 1) Z logičnimi enačbami implementirajte 4-bitno vezje, katerega delovanje določata signala `count in up` na sledeč način:

<code>count \ up</code>	0	1
0	shift left	shift right
1	count down	count up

Vezje naj ima tudi sinhroni `reset` signal, ki postavi vsebino števca na 0.

## Naloga (2)

2) Števec omejite tako, da bo štel po modulu 10.

Pomoč: izven procesa definiramo signal, ki je enak 1, ko je števec  $\geq 9$  (sig\_up) in signal, ki je enak 1, ko je števec  $\geq 10$  ali 0 (sig\_down). Prvi signal nam pove, da se mora števec ob štetju navzgor postaviti na 0, drugi pa da se mora števec ob štetju navzdol postaviti na 9.

Signala vključimo v enačbo v procesu na naslednji način:

```
q(i) = (      up AND      sig_up  AND 0) OR
        (      up AND (NOT sig_up) AND (enacbe_za_stetje_gor)) OR
        (not up AND      sig_down AND (i-ti bit v številki 9)) OR
        (not up AND (NOT sig_down) AND (enacbe_za_stetje_dol));
```