## 3. vaja: Utripanje LED na razvojni plošči

Razvili bomo vezje, ki krmili svetleče diode na razvojni plošči DEO-Nano. Najprej bomo na LED prikazovali stanje števca, na koncu bomo pa svetilnost LED določali s pulzno-širinskim modulatorjem. Model vezja bomo naredili v orodju SHDL, generirali VHDL in ga vključili v vnaprej pripravljen projekt orodja Quartus Prime.

## Števec v orodju SHDL

- Naredi model vezja z imenom stevec, ki ima 8-bitni izhod led. V vezju naj bo 30-bitni nepredznačen vektor, ki se povečuje ob uri. Zgornjih 8 bitov naj bo povezanih na led. Na simulaciji poskusi delovanje števca, ki naj se ob vsakem ciklu ure poveča za en milijon (približno 2<sup>20</sup>).
  - Koliko ciklov potrebuje števec, da prešteje do konca in naredi preliv?
- 2. Popravi števec tako, da bo pri uri frekvence 50 MHz naredil preliv v približno eni sekundi. Kakšen mora biti v tem primeru inkrement števca?

## Projekt za razvojno ploščo

- 3. Prenesi in razširi datoteke vzorčnega projekta <u>stevec.zip</u> za program Quartus v svojo mapo:
  - stevec.qpf (Quartus Project File) je glavna datoteka s katero odpremo projekt
  - **stevec.qsf** vsebuje nastavitve projekta, izbiro vezja Cyclone IV E EP4CE22F17C6, definicije priključkov in vključenih datotek
  - de0.sdc (Synopsys Design Constraint) določa časovne parametre pri prevajanju vezja
  - stevec.vhd je datoteka z opisom vezja v jeziku VHDL

V nastavitvah projekta (*Assignments, Settings*) je določeno ime glavne datoteke, v kateri mora biti istoimenski model vezja, v našem primeru: **entity** stevec. Zunanji signali stavka **port** bodo po prevajanju povezani na priključke razvojne plošče. Povezave so določene v tekstovni datoteki stevec.qsf. Če bomo v modelu vezja določili druge zunanje signale, jih bo prevajalnik razmestil na naključne povezave plošče, kar lahko naredi kakšen kratek stik !



4. Odpri Quartus Prime z dvoklikom na **stevec.qpf**. V programu odpri glavno datoteko v urejevalniku VHDL kode. Zamenjaj vsebino s kodo števca, ki jo generira orodje SHDL, vezje prevedi in ga naloži na razvojno ploščo (odpri Programmer, klikni na Start). Opazuj delovanje števca na svetlečih diodah.

Hardware Setup USB-Blaster [USB-0] Mode: JTAG					
Enable real-time ISP to allow background programming when available					
▶ <sup>™</sup> Start	File	Device	Checksum	Usercode	Program/ Configure
■ <sup>™</sup> Stop	stevec.sof	EP4CE22F17	001349A8	001349A8	

## Pulzno širinski modulator

Če LED utripajo zelo hitro, ne bomo zaznali utripanja, ampak le nižjo svetlost. Svetlost LED uravnavamo s širino impulzov oz. razmerjem časa logične 1 in logične 0.

- V orodju SHDL pripravi vezje z imenom pwm. Vhod je 4-bitni nepredznačen vektor data, izhod pa enobitni pulz. V vezju je notranji 4-bitni števec in primerjalnik: pulz naj bo 1, ko je števec manjši od vhodne vrednosti data. Preveri delovanje s simulacijo in prenesi VHDL kodo v Quartus.
- 2. V Quartusu dodaj novo datoteko (*New VHDL file*) v katero kopiraj generirano VHDL kodo iz orodja SHDL. Datoteko shrani kot **pwm.vhd** in vključi v projekt (*Project, Add Current File to Project*).
- 3. V glavni datoteki (stevec.vhd) je potrebno povezati **pwm** (s stavkom **port map**). Podatkovni vhod naj bo vezan na zgornje štiri bite 30-bitnega števca. Izhod naj bo vezan na enobitni notranji signal, ki ga nato poveži na vse bite vektorja **led**. Preizkusi delovanje na razvojni plošči.
  - Primer uporabe stavka port map najdeš v orodju SHDL, če namesto VHDL odpreš Test Bench. Vsi signali na desni strani puščice (=>) morajo biti deklarirani v glavni datoteki.

Izhod pwm je enobiten signal. Za povezavo na izhod led lahko napišeš stavek, ki osem enobitnih signalov sestavi v izhodni vektor (operator &) ali pa stavek, ki preveri vrednost izhoda in s pogojno prireditvijo krmili led. Pazi na narekovanje v jeziku VHDL (če je izhod '1', naj bo led "11111111").