

7. vaja: Skaliranje podatkov z nasičenjem

Naredili bomo komponento vezja, ki izvaja skaliranje podatkov iz A/D pretvornika na merilni platformi Red Pitaya. Uporabili bomo poenostavljen vmesnik za povezavo s procesorjem.

1. Naredi nov projekt za razvojno ploščo RedPitaya in v projekt vključi datoteko `rtl/red_pitaya_proc.vhd` iz paketa: [RedPitaya2017.zip](#)

- o **clk_i, rstn_i**: ura in reset signal (aktivna 0)
- o **sys_addr**: 32-bitni naslovni vhod
- o **sys_wdata**: 32-bitni vhodni podatek
- o **sys_wen, sys_ren**: vhodna signala za branje in pisanje na vodilo
- o **adc_i**: 14-bitni vhodni merilni podatek
- o **sys_rdata**: 32-bitni izhodni podatek
- o **sys_err, sys_ack**: izhoda za javljanje napake in potrditev prenosa
- o **adc_o**: 14-bitni skaliran merilni izhod



2. V komponenti je 8-bitni register **reg** v katerem je faktor za skaliranje 14-bitnega vhodnega signala **adc_i**. Signal **adc_i** najprej pretvori v notranji signal tipa *signed*. Vrednost iz 8-bitnega **reg** pretvori v 9-bitno predznačeno število in pomnoži z **adc_i**. Rezultat skrči na 14-bitni vektor in pelji na izhod **adc_o**. Če pride pri množenju do prekoračitve območja 14-bitnih vrednosti, naj gre izhod v pozitivno ali negativno nasičenje.
3. Dodaj v opis vezja logiko pisanja na sistemsko vodilo, ki nastavlja vrednost **reg**. Dodaj sinhroni proces (`rising_edge(clk_i)`), ki ob **rstn_i='0'** postavi register na 0, ob **sys_wen='1'** in naslovu **sys_addr(19 downto 0)=X"00000"** pa prenese v register spodnjih 8 bitov iz vodila **sys_wdata**.
4. Preveri delovanje komponente za skaliranje podatkov s testno strukturo: [TestProc.vhd](#). Testna struktura zapiše v register vrednost 5, čez nekaj časa pa vrednost 9, na vhod **adc_i** pa pripelje vzorce sinusnega signala amplitude **1540**. Amplituda je izbrana tako, da ostanejo vrednosti po množenju s 5 znotraj območja (-8192, 8191), po množenju z 9 pa so nekatere izven območja in na simulaciji lahko opazujemo učinek množenja z nasičenjem.

V simulatorju nastavi signale **adc_i** in **adc_o** kot predznačene (desni klik **Radix, Signed Decimal**), nato pa kot analogne (**Waveform Style, Analog**)

