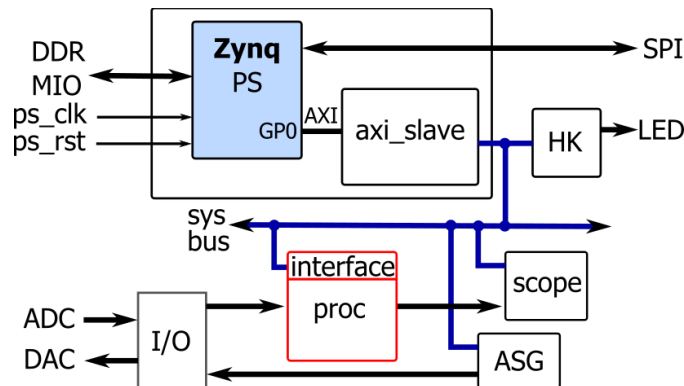


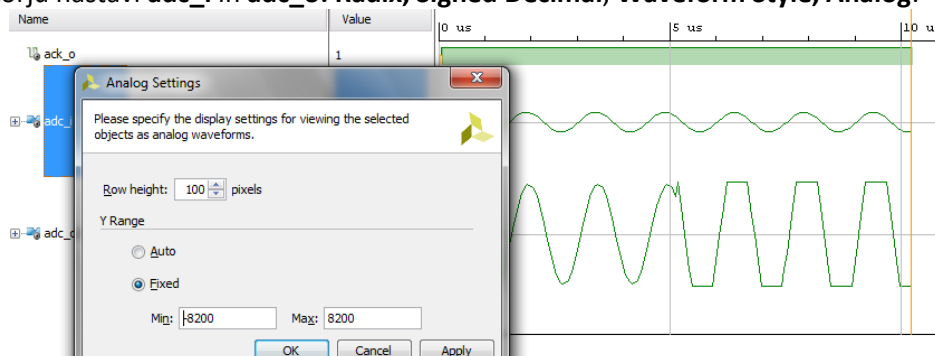
9. vaja: Skaliranje podatkov na Red Pitayi

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.



- Odpakiraj [RedPitaya2017.zip](#), naredi nov projekt za razvojno ploščo RedPitaya in v projekt vključi datoteko `rtl/red_pitaya_proc.vhd` s priključki:
 - clk_i, rstn_i**: ura in reset signal (aktivna 0)
 - addr_i**: 32-bitni naslovni vhod
 - wdata_i**: 32-bitni vhodni podatek
 - wen_i, ren_i**: vhodna signala za branje in pisanje na vodilo
 - adc_i**: 14-bitni vhodni merilni podatek
 - rdata_o**: 32-bitni izhodni podatek
 - err_o, ack_o**: izhoda za javljanje napake in potrditev prenosa
 - adc_o**: 14-bitni skaliran merilni izhod
- V vezju je 8-bitni register **reg** v katerem je faktor za skaliranje vrednosti iz **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.
- Dodaj v opis vezja logiko pisanja na sistemsko vodilo za nastavitve registra **reg**. Napiši sinhroni proces, ki ob **rstn_i='0'** postavi register na 0, ob **wen_i='1'** in naslovu **addr_i(19 downto 0)=X"00000"** pa prenese v register spodnjih 8 bitov iz vodila **wdata_i**.
- Preveri delovanje vezja 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 **adc_i** in **adc_o**: Radix, Signed Decimal, Waveform Style, Analog:



10. vaja: Preizkus na plošči

1. Uporabi projekt iz prejšnje vaje v katerega naj bodo vključene vse podmape (rtl, sdc in tcl). V spodnjem oknu odpri zavihek Tcl Console in izvedi ukaze:
 - premakni se v trenutno mapo, npr: `cd c:/proj/divs/redpitaya`
 - izvedi skripto, ki naredi blokovni diagram: `source tcl/system.tcl`
 - in skripto za vključitev datotek: `source tcl/files.tcl`
2. Preveri datoteke projekta in se prepričaj, da vsebuje zadnjo verzijo datoteke **red_pitaya_proc.vhd**, nato pa izvedi prevajanje (sintezo, implementacijo in Generate Bitstream).
 - Program javi nekaj kritičnih opozoril (false path constraint), ki jih lahko ignoriramo.
 - Rezultat prevajanja je datoteka **red_pitaya_top.bit** v podmapi <ime_projekta>.runs.
3. Preizkusi delovanje na Red Pitayi. V laboratoriju LRNV bomo Red Pitayo priklopili kar neposredno na računalnikovo drugo omrežno kartico, ki jo nastavimo na lokalni naslov: **192.168.1.101**. Red Pitaya bo dostopna na naslovu **192.168.1.15** v konzoli SSH (ime: **root**, geslo: **root**). Za delo s konzolo uporabi program **PuTTY**, za prenos datotek pa [WinSCP](#).
 - Prenesi datoteko: **red_pitaya_top.bit** v lokalno mapo na Red Pitayo (npr. root) in jo nato v konzoli kopiraj v /opt/redpitaya/fpga/. Če sistem ne dovoli kopiranja, najprej izvedi ukaz **rw**.
 - V spletni brskalnik napiši naslov: **192.168.1.15** in poženi aplikacijo Oscilloscope. Strežnik je nastavljen tako, da bo v FPGA naložil datoteko **red_pitaya_top.bit**.

Nastavi nek signal na izhod signalnega generatorja (OUT1) in poveži izhod z vhodom osciloskopa. Prikazani signal bo skaliran s faktorjem, ki ga določa vrednost registra. Register nastavljamo v konzoli s programom monitor, tako da podamo naslov in vrednost, npr: `monitor 0x40500000 10`