

# 7. Vaja: Skaliranje signala

Naredili bomo logično vezje za skaliranje amplitude vhodnega signala z nasičenjem

## Množenje s pomikanjem in seštevanjem

Skaliranje digitalnega signala naredimo z operacijo množenja. Poglejmo si, kako naredimo množenje predznačenih števil z 2. Množenje dvojiškega števila z 2 predstavlja dodajanje ene ničle na desni strani (podobno kot množenje desetiškega števila z 10):

$$\begin{aligned}
 5 \times 2 &= 10 & 00101 \times 10 &= 001010 \\
 -5 \times 2 &= -10 & 11011 \times 10 &= 110110
 \end{aligned}$$

Množenje z 10 pretvorimo v vsoto dveh pomaknjenih vrednosti: produkta z 8 in z 2:

$$\begin{aligned}
 5 \times 10 &= 5 \times 8 + 5 \times 2 & 00101 \times 1000 &= 00101000 \\
 & & 00101 \times 0010 &= \underline{00001010} \\
 & & & 00110010
 \end{aligned}$$

Če vzamemo najvišjo pozitivno 5 bitno vrednost ( $01111=15_{10}$ ) in jo množimo z 10, moramo za rezultat rezervirati 9 bitov:  $0\ 1001\ 0110 = 150_{10}$ . V splošnem ima rezultat množenja z 10 štiri dodatne bite.

## Vezje za skaliranje in nasičenje

V orodju [SHDL](#) opiši vezje, množi 14-bitni vhodni signal **din** s faktorjem 2 ali 10. Na vhodu naj bo še 32-bitni parameter **set**, ki določa faktor skaliranja:

- set=0, dout = din
- set=1, dout=din x 2
- set=2, dout=din x 10

Izhodni signal **dout** naj bo prav tako 14-biten. Kadar ima vhod veliko vrednost, bi potrebovali po množenju za zapis izhoda več bitov, sicer pride do preliva vrednosti. Namesto preliva, bomo naredili logiko, ki postavi izhod na zgornjo ali spodnjo mejo nasičenja.

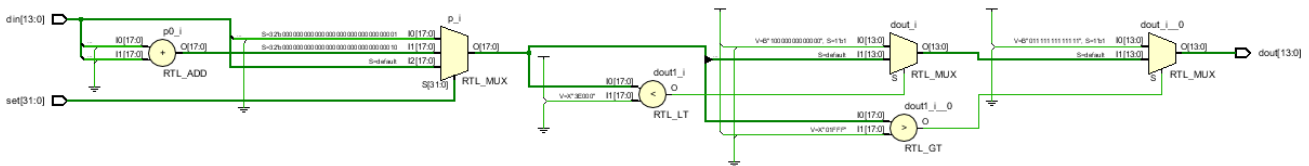
1. V orodju SHDL opiši vezje z vhodoma **din** in **set** ter izhodom **dout**. Deklariraj še 18 biten notranji signal p, ki ga potrebujemo za vmesni produkt in opiši logiko za določanje produkta glede na set. Namesto množenja uporabi dodajanje ničel in seštevanje. Preveri delovanje v simulatorju.
2. Ugotovi, kakšna sta zgornja in spodnja meja nasičenja 14-bitnih vrednosti. Dodaj logiko, ki pretvori notranji signal v 14-biten izhod, kadar je znotraj meja, ali pa postavi izhod v nasičenje.

|      | 1 |     |      |      |      |      |       |       |       |       | 2 |      |      |      |      |       |       |       |      |      |
|------|---|-----|------|------|------|------|-------|-------|-------|-------|---|------|------|------|------|-------|-------|-------|------|------|
| set  |   |     |      |      |      |      |       |       |       |       |   |      |      |      |      |       |       |       |      |      |
| din  | 0 | 200 | 600  | 800  | 1000 | 8000 | -8000 | -1000 | -800  | -600  | 0 | 200  | 600  | 800  | 1000 | 8000  | -8000 | -1000 | -800 | -600 |
| dout | 0 | 400 | 1200 | 1600 | 2000 | 8191 | -8192 | -2000 | -1600 | -1200 | 0 | 2000 | 6000 | 8000 | 8191 | -8192 | -8000 | -6000 |      |      |

## Sinteza vezja in prevajanje projekta

Spletno orodje ob prevajanju generira vezje v standardnem jeziku VHDL, ki ga bomo vključili v obstoječ projekt za razvojno ploščo Red Pitaya. Projekt je v mapi: C:/UKE/vaja7.

3. V orodju Vivado odpre projekt ClassicBD in dvoklikom na proc(RTL) datoteko z opisom komponente za skaliranje signalov. Zamenjaj vsebino z VHDL kodo, ki jo generira spletno orodje in potrdi osvežitev projekta (Refresh Changed Modules).
4. Določi komponento proc(RTL) kot glavno (desni klik, Set as Top) in naredi elaboracijo: RTL ANALYSIS, Open Elaborated Design, ki prikaže logične gradnike v modelu komponente:



5. Pred prevajanjem projekta ponastavi glavno komponento ClassicBD\_wrapper (Set as Top), nato pa izberi vse korake prevajanja: Flow, Generate Bitstream.

## Preizkus na razvojni plošči

6. Na razvojni plošči Red Pitaya poveži izhod OUT2 z vhodom IN1. Prikluči ploščo na računalnik in se poveži s programom SSH Client. Odpre raziskovalec za prenos datotek, izberi lokalno mapo: C:\UKE\vaja7\ClassicBD.runs\impl\_1 in prenese na razvojno ploščo datoteko:

### ClassicBD\_wrapper.bit

7. V spletnem brskalniku odpre stran: 192.168.1.15 in aplikacijo osciloskop. Nastavi na OUT2 sinusni signal, ki se mora pojaviti tudi na IN1. Spreminjaj nastavitve set z vpisom ukaza v terminal in opazuj izhod:

```
monitor 0x40300050 1
```

```
monitor 0x40300050 2
```

