

### 3. laboratorijas darbs – sinusoīdu ģenerēšana un filtrēšana

2020. gada pavasara semestris

#### Ievads

3. laboratorijas darbs – ir paredzēts kursā apgūto teorētisko un praktisko zināšanu pielietošanu īstenojot ciparu signāla ģenerāciju izmantojot DDS, un filtrāciju, izmantojot FIR un IIR filtrus.

Galvenais laboratorijas darba mērķis ir panākt, ka tiek ar ciparu filtriem atdalītas divas sinusoīdas.

#### Sistēmas Parametri

$f_s$  (ciparu daļas sample clock) = 22050 (audio standarts).

Rst – active high. Pēc rst sistēma uzsāk darboties no nulles (visas vērtības un starprezultāti tiek nonullēti).

Fiksētā punkta daļas sistēmas bitu platums – signed 16b (bitu platumi dziļākos moduļos IIR filtram atšķirsies).

$$f_1 = 32 * k$$

$$f_2 = 48 * k$$

k – studenta apliecības pēdējie 2 cipari.

Piemēram, ja jūsu studentu apliecības pēdējie 2 cipari ir "09", tad  $f_1 = 32 * 09 = 288$  Hz,  $f_2 = 48 * 09 = 432$  Hz.

#### Laboratorijas darba uzdevumi:

1. Veidot laboratorijas darba gaitas aprakstu – atskaiti. Noformēt atbilstoši, iekļaut dažādu uzdevumu rezultātu shēmas, laika diagrammas no Simulink, Modelsim, utml. – atskaite ir obligāta, lai iegūtu jebkādu atzīmi.
2. MATLAB-Simulink peldošā punkta daļa, jeb izveidot references filtru, pret ko varēs salīdzināt tālākos lab.darba uzdevumu rezultātus) - izveidot attēlā redzamo sistēmu, kas sastāv no diviem sinusoidālas formas signāla ģeneratoriem ar frekvencēm  $f_1$  un  $f_2$ , summatora un diviem joslas filtriem (viens – FIR, otrs – IIR). *Ieteikums ir veiksmīgi izpildītu 2. uzdevumu ar Save As saglabāt, lai tas paliek strādājošs, un ar šo rezultātu var pārējos uzdevumus salīdzināt jebkurā brīdī.*
  - a. Izvietot divus sinusoidālas formas signāla ģeneratorus un to rezultātu sasummēt – izmantot iebūvētos Simulink blokus.
  - b. Izveidot FIR bandpass filtru, izmantojot Simulink esošo bloku [Discrete FIR Filter](#). Koeficienti jāaprēķina izmantojot MATLAB funkciju [fir1\(\)](#). *Ieteikums - ar MATLAB funkciju [freqz\(\)](#) var apskatīties vizuāli filtra raksturliķnes.* Filtra kārtu (joslas platumu) izvēlēties, lai filtra  $f_1$  vājinājums pie frekvences  $f_2$  būtu vismaz - 20dB (un otrādi), pārbaudīt ar [freqz\(\)](#) un iekļaut atskaitē.

- c. Izveidot IIR bandpass filtru, izmantojot Simulink esošo bloku [Discrete Filter](#). Koeficienti jāaprēķina izmantojot MATLAB funkciju [iirpeak\(\)](#) - iegūst koeficientus 2. kārtas IIR filtram ar gan rekursīvo, gan nerekursīvo daļu (kopā 6 koeficienti, no kura vienam jābūt "1") Kā iepriekš, ar [freqz\(\)](#) var apskatīties filtra raksturliķni. Filtra raksturliķni iekļaut atskaitē. Pārbaudīt abus 2-b, 2-c izstrādātos filtrus darbībā, ka no sinusoīdas summas pareizi tiek izfiltrētas abas sinusoīdas. Rezultātus iekļaut atskaitē.
  - d. Aizstāt IIR bandpass filtra bloku Discrete Filter ar manuāli izveidotu filtru no elementārajiem elementiem - aizturēm, reizinātājiem un summatoriem, pārbaudīt, ka strādā bandpass filtrēšana. Filtru struktūrshēmu izvēlēties izmantojot MATLAB [dfilt](#) [df1](#) vai [df2](#).
  - e. Pārbaudīt, ka vizuāli redzams, ka sinusoīdas tiek izfiltrētas, un rezultāts ir tuvs tam, kas 2-c punktā. Rezultātu iekļaut atskaitē. Pareizi strādājošs, šis ir jūsu references filtrs. Pirms nākamā uzdevuma sākšanas, saglabājiet šo ar Save As, jo noteikti vajadzēs salīdzināt.
3. MATLAB-Simulink fiksētā punkta daļa – pārveidot 2. uzdevumā izstrādāto shēmu uz fiksēto punktu.
    - a. Blokhēmā blokiem jāuzstāda fiksētā punkta parametri. Izmantojiet fiksētā punkta struktūru 2.14, jeb s1.14 (divi biti pirms punkta, no kuriem viens ir zīmes bits, un 14 biti pēc punkta). Var pārveidot katra bloka ieejas/izejas veidus, vai lietot Convert bloku. IIR bandpass filtra modulī, papildus aizturēm, reizinātājiem un summatoriem atļauts izmantot arī Convert bloku, ja vēlaties).
    - b. IIR Filtra starprezultātu bitu platums var būt atkarīgs no koeficientiem. Pielāgot filtra starprezultātu bitu platumu, lai iegūtu strādājošu IIR filtru. Ieteikums – sekot līdzi un atcerēties no lekcijas kā novietošanas binārais punkts, atkarībā no darbības veida (summa, reizinājums), kā arī vai nenotiek bufera pārpilde (overflow).
    - c. Pārbaudīt, ka vizuāli redzams, ka sinusoīdas tiek atdalītas, līdzīgi kā 2. uzdevumā.
  4. VHDL daļa.
    - a. Izveidot VHDL moduli - parametrizējamu sinusoidāla sprieguma ģeneratoru. Ģenerētā signāla frekvencei jābūt parametrizējamai (vai nu ar generic, vai port ievadu), ar frekvenci, ko var uzstādīt, kad šis bloks tiks izsaukts. Izejas bitu platumu 16, nosaukums sin\_gen (pamats dots 4. lekcijā).
    - b. Izveidot sin\_gen testēšanas benķi testbench "sin\_gen\_tb.vhdl", lai pārlielinātu par tā darbības

- pareizību. Atskaitē iekļaut Modelsim Analog veida waveformas. Jāpārlicinās par parametrizējamās frekvences pareizību. Frekvence blokam jāievada vai nu hercos, vai normalizētajā frekvencē pret  $f_s$ .
- Izveidot moduli "sin\_gen\_sum", kurā iekļauti divi sin\_gen ģeneratori ar frekvencēm  $f_1$  un  $f_2$ , un sasummēts to rezultāts (tas būs ieejas signāls priekš filtru bloka). Izveidot testēšanas beņķi "sin\_gen\_sum\_tb.vhdl" un pārbaudīt vai strādā summa pareizi. Atskaitē iekļaut Modelsim Analog waveformas.
  - Izveidot FIR joslas filtru, kas noregulēts uz  $f_1$ , izmantojot Altera Megawizard plugin IP Core ģeneratoru. Filtra kārtā, koeficienti, frekvences jāizmanto tās pašas, kas 2-b, 3-a.
  - Iekļaut projektā IIR joslas filtra VHDL failu, kas iegūts izmantojot MATLAB-SIMULINK HDL Coder. Šādi no 3-b īstenotā filtra tiks uzģenerēts VHDL kods.
  - Īstenot attēlā redzamo elementu savienojumu VHDL failā lab3.vhdl
    - Divi sin\_gen, viens uzstādīts ar  $f_1$ , viens ar  $f_2$
    - Sin\_gen izeju summēšana vienā signālā
    - Summētā signāla filtrēšana divos atsevišķos signālos
  - Izveidot kopējā projekta testēšanas beņķi ""lab3\_tb.vhdl" testbench, lai pārlicinātos par pareizu darbību, salīdzinot ar 1. un 2. uzdevumā paveikto. Atskaitē iekļaut Modelsim Analog waveformas.

- Sin\_gen\_sum\_tb.vhdl
- FIR filtra megawizard plugin uzģenerētie faili
- Ar Matlab HDL Coder uzģenerētais IIR filtra .vhdl fails.
- Top-level projekta fails lab3.vhdl.

Laboratorijas darba failos NEVAJAG iesniegt simulāciju rezultātu failus, jo tie tipiski ir ļoti lieli. Pārbaudot darbu mācībspēki tos paši uzģenerēs.

Laboratorijas darbs jāiesniedz, to augšuplādējot kursa ORTUS šī uzdevuma sadaļā.

### Laboratorijas darba vērtējums

Šis laboratorijas darbs tiks vērtēts ar atzīmi, kas sastādīs 7 balles no visa kursa vērtējuma (kursa sākumā bija plānots veidot individuālos projektus, taču ņemot vērā ārkārtējās situācijas apstākļus, pieeja tika mainīta). Atlikušās 3 balles sastādīs vērtējums, kas iegūts no jau izpildītajiem patstāvīgajiem uzdevumiem.

Vērtējums uzdevumiem:

Uzdevums	Vērtējums
1	1
2	1
3	2
4	3
4-a,b,c	1
4-d,e	1
4-f,g	1
<b>kopā</b>	<b>7</b>

### Laboratorijas darba iesniegšana

Jāiesniedz:

- Laboratorijas darba atskaite .docx vai .pdf vai citos formātos
- Visi Simulink faili
- Visi VHDL faili un FPGA projekts
  - Sin\_gen.vhdl
  - Sin\_gen\_tb.vhdl – testēšanas beņķis
  - Sin\_gen\_sum.vhdl

### Ieteikums

Ja sanāk uz kādu uzdevuma punktu "iesēsties" vairāk kā 1h, tad lūdzu jautāt konsultāciju mācībspēkiem vai studentiem-kolēģiem.

Veiksmi !

