Summary - Complex Mixer

Name	complex_mixer
Worker Type	Application
Version	v1.2
Release Date	August 2017
Component Library	ocpi.training.components
Workers	complex_mixer.hdl complex_mixer.rcc
Tested Platforms	c7-x86_64, linux-x13_3-arm, xsim, Matchstiq-Z1(PL)(Vivado 2017.1)

Functionality

The Complex Mixer consists of a Numerically Controlled Oscillator (NCO) and a complex multiplier. Complex IQ data is received on the input port and is multiplied with the output of the NCO and put on the output port.

Worker Implementation Details

complex_mixer.hdl

Figure 1 diagrams the complex mixer.

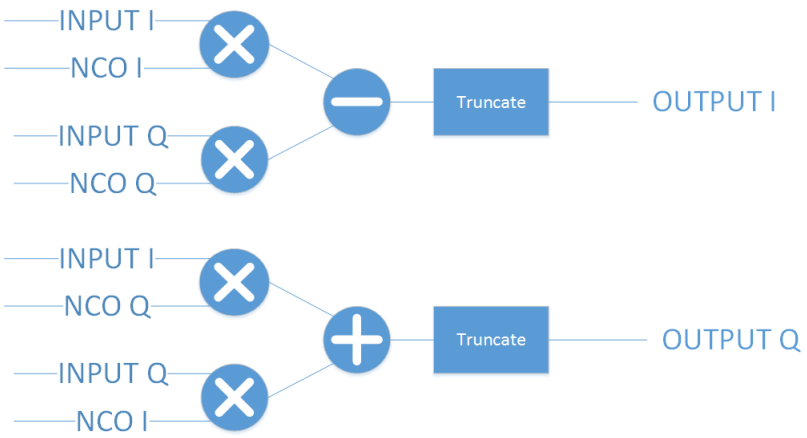

Figure 1: Complex Mixer Functional Diagram

The control properties for all implementations provide the ability to *enable*(or bypass) the worker and set the tune frequency via *phs_inc*.

The HDL worker leverages Xilinx Vivado cores to implement the NCO (DDS Compiler v6.0.13) and complex multiply (Complex Multiplier v6.0.12) functions. The Vivado IP Catalog tool was used to generate both cores. For both cores, their respective default configuration settings were chosen with the following exceptions:

- DDS Compiler 6.0.13 (dds_compiler.xci)

  Parameter Selection = Hardware Parameters

  Output Width = 16

  Phase Increment Programmabililty = Streaming

  Optional Pins: Has Phase Out is unchecked and ARESETn checked

- Complex Multiplier v6.0.12 (complex_multiplier.xci)

  Control Signals: ACLKEN and ARESETn checked

The dds_compliler v6.0.13 implements a phase generator and sin/cos look-up tables (LUT), and has a latency of 7. The complex_multiplier v6.0.12 uses 3 DSP48E1s and has a latency of 6.

In conjunction with the *enable*(or bypass) control property, the HDL worker provides a the ability to select different (*data_select*) data to output in bypass mode: input or output of the NCO.

complex_mixer.rcc

The RCC worker leverages liquid-dsp v1.2 and its *nco* class to generate the internal NCO used in the algorithm. More information on this liquid-dsp module can be seen in the online documentation: liquid-dsp.
In the RCC version of this component the samples are converted from fixed point to floating point numbers in order to do that math on a GPP. This conversion introduces a small amount of error in the output data and should be accounted for when it is used in an application. The conversion equations are as follows:

$$iq\_float = \frac{iq\_fixed}{2^{15} - 1} \tag{1}$$

$$iq\_fixed = iq\_float * (2^{15} - 1) \tag{2}$$

In the RCC worker a conversion needs to be done for the phase increment to adhere to the way the HDL phase increment is implemented. The conversion was done in the RCC version of this component because the division operation is very resource intensive in HDL. The conversion from the component property to the liquid-dsp interface input property is as follows:

$$liquid\_phs\_inc = phs\_inc * \frac{2\pi}{0x7FFF * 2} \tag{3}$$

Theory

The Complex Mixer worker inputs complex signed samples and performs a complex multiply with a digital sine wave produced by an numerically controlled oscillator (NCO). The resulting output data is a frequency shifted version of the input data.

The magnitude of the frequency shift is determined by the output frequency of the NCO, which can be calculated with the following equation:

$$nco\_output\_freq = sample\_freq * \frac{phs\_inc}{2^{phs\_acc\_width}} \tag{4}$$

In this component, `phs_inc` is runtime configurable and has a data type of 16 bit signed short. `phs_acc_width` is fixed at 16. The input clock frequency is the sample rate of the samples. The amplitude of the NCO's sine wave is fixed at the full range of a signed 16 bit value.

Block Diagrams

Top level

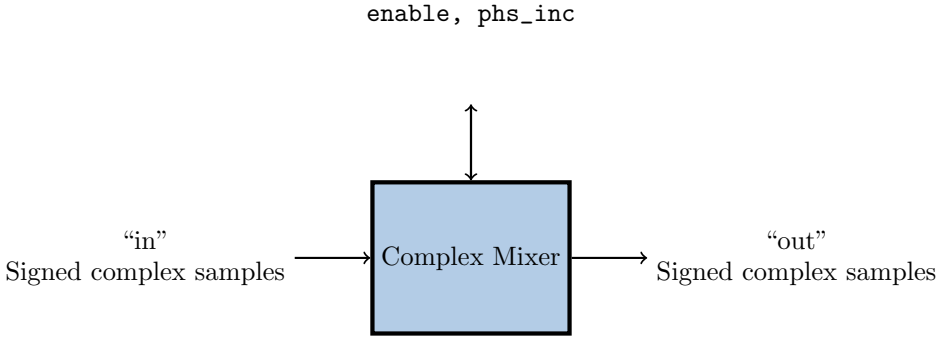

Figure 2: Complex Mixer Top Level Block Diagram

State Machine

Not Applicable

Source Dependencies

complex_mixer.rcc

- Liquid DSP (provided as `av-prereq-liquid-*.rpm`)

complex_mixer.hdl

- training_project/components/complex_mixer.hdl/complex_mixer.vhd
- training_project/components/complex_mixer.hdl/vivado_ip/complex_multiplier_stub.vhd
- training_project/components/complex_mixer.hdl/vivado_ip/complex_multiplier_sim_net.vhd
- training_project/components/complex_mixer.hdl/vivado_ip/complex_multiplier.vho
- training_project/components/complex_mixer.hdl/vivado_ip/complex_multiplier.xci
- training_project/components/complex_mixer.hdl/vivado_ip/complex_multiplier.edf
- training_project/components/complex_mixer.hdl/vivado_ip/dds_compiler_stub.vhd
- training_project/components/complex_mixer.hdl/vivado_ip/dds_compiler_sim_net.vhd
- training_project/components/complex_mixer.hdl/vivado_ip/dds_compiler.vho
- training_project/components/complex_mixer.hdl/vivado_ip/dds_compiler.xci
- training_project/components/complex_mixer.hdl/vivado_ip/dds_compiler.edf

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
`enable`	Bool	-	-	Readable, Writable	Standard	true	Enable(true) or bypass(false) mixer

`phs_inc`	Short	-	-	Readable, Writable	*	-8192	Phase increment of NCO
							
							`* -2^(NCO_DATA_WIDTH_p-1)`
							`to +2^(NCO_DATA_WIDTH_p-1)-1`

Worker Properties

complex_mixer.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	`data_select`	Bool	-	-	Readable, Writable	Standard	false	In Bypass Mode: selects data to output: 0=input data, 1=output of NCO

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
in	false	iqstream_protocol	false	-	Signed complex samples
out	true	iqstream_protocol	false	-	Signed complex samples

Worker Interfaces

complex_mixer.hdl

Type	Name	DataWidth	Advanced	Usage
StreamInterface	in	32	-	Signed Complex Samples
Type	Name	DataWidth	Advanced	Usage
StreamInterface	out	32	-	Signed Complex Samples

Control Timing and Signals

The Complex Mixer HDL worker uses the clock from the Control Plane and standard Control Plane signals.

There is a startup delay for this worker. Once the input is ready and valid and the output is ready, there is a 6 clock cycle start delay (pipeline delay of complex multiplier). After this startup delay, valid output data is given 6 clock cycles after input data is taken.

Latency
6 clock cycles

Performance and Resource Utilization

RCC

Needs to be tested to populate this section.

HDL

Device	Registers	LUTs	Fmax	Memory/Special Functions	Design Suite
Zynq XC7Z020-1-CLG484	450	252	307.882 MHz	DSP48E1 = 3	Vivado 2017.1

Test and Verification

Test cases are derived from the number of properties, and their respective values, as listed in the complex_mixer-test.xml. Specifically, the complex_mixer.rcc and complex_mixer.hdl implementations tested, as follows:

1) Bypass (RCC & HDL): The input data is forwarded to the output port. For verification of this case, the output file is byte-wise compared to the input file.

2) Normal mode (RCC & HDL): The NCO is configured to tune the input signal to baseband. For verification, an FFT of the output data is performed and the max value of the FFT is checked to be at DC (0 Hz).

For all cases, the input file contains a tone of 12.5 Hz sampled at 100 Hz and an amplitude of 32767.