Summary - Complex Mixer

Name	complex_mixer
Worker Type	Application
Version	v1.5
Release Date	4/2019
Component Library	ocpi.assets.dsp_comps
Workers	complex_mixer.hdl complex_mixer.rcc
Tested Platforms	xsim, isim, modelsim, alst4, ml605, ZedBoard(PL), Matchstiq-Z1(PL),

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.

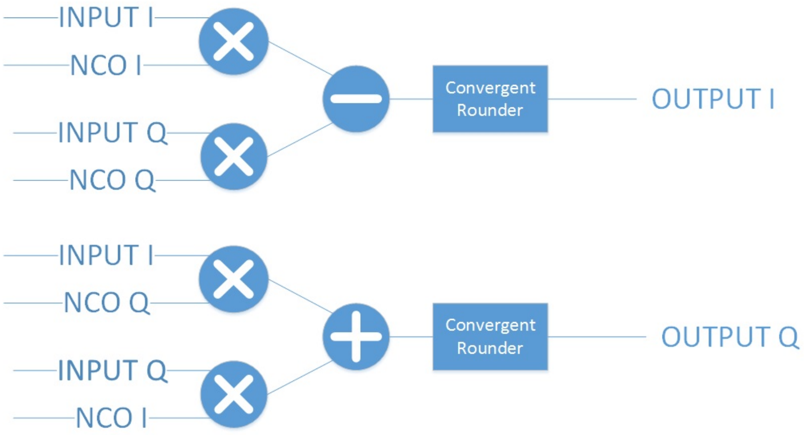

Figure 1: Complex Mixer Functional Diagram

Build time parameters can be used to control the width of the NCO data output, the width of the input data, and the number of stages in the Coordinate Rotation Digital Computer (CORDIC) used to implement the NCO. Additionally, there is a parameter to control insertion of a peak detection circuit.

An `enable` input is available to either enable (true) or bypass (false) the circuit. In bypass mode, pipe-lining registers are not used. FPGA multipliers are used to process input data at the full clock rate. This worker will produce valid output two clock cycles after each valid input.

complex_mixer.rcc

The RCC worker leverages liquid-dsp v1.2 and its *nco* class to generate the internal NCO used in the algorithm. OpenCPI provides RPMs for installing liquid-dsp, which must be installed in order to build and run this worker. 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 also runtime configurable via the `mag` property. Note that the `mag` property value should only ever be set to a value within the following range in order for the worker to operate properly.

$$-2^{(NCO\_DATA\_WIDTH\_p-1)} <= mag <= 2^{(NCO\_DATA\_WIDTH\_p-1)} - 1 \tag{5}$$

A positive and negative `phs_inc` will mix up and down, respectively. The following equation can be used as an aid for setting the `phs_inc` to have the desired mixing affect.

$$x_{out}[n] = x_{in}[n] * \frac{mag}{2^{NCO\_DATA\_WIDTH\_p-1}} * e^{\left(j2\pi\left(sample\_freq*\frac{phs\_inc\ *\ n}{2^{phs\_acc\_width}}\right)+phs\_init\right)} \quad \forall\ n,\ n \geq 0 \tag{6}$$

Block Diagrams

Top level

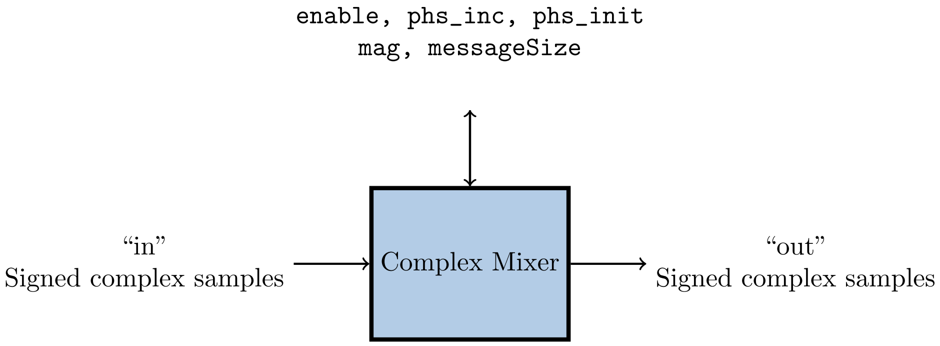

Figure 2: Complex Mixer Top Level Block Diagram

State Machine

Only one finite-state machine (FSM) is implemented by this worker. The FSM supports Zero-Length Messages.

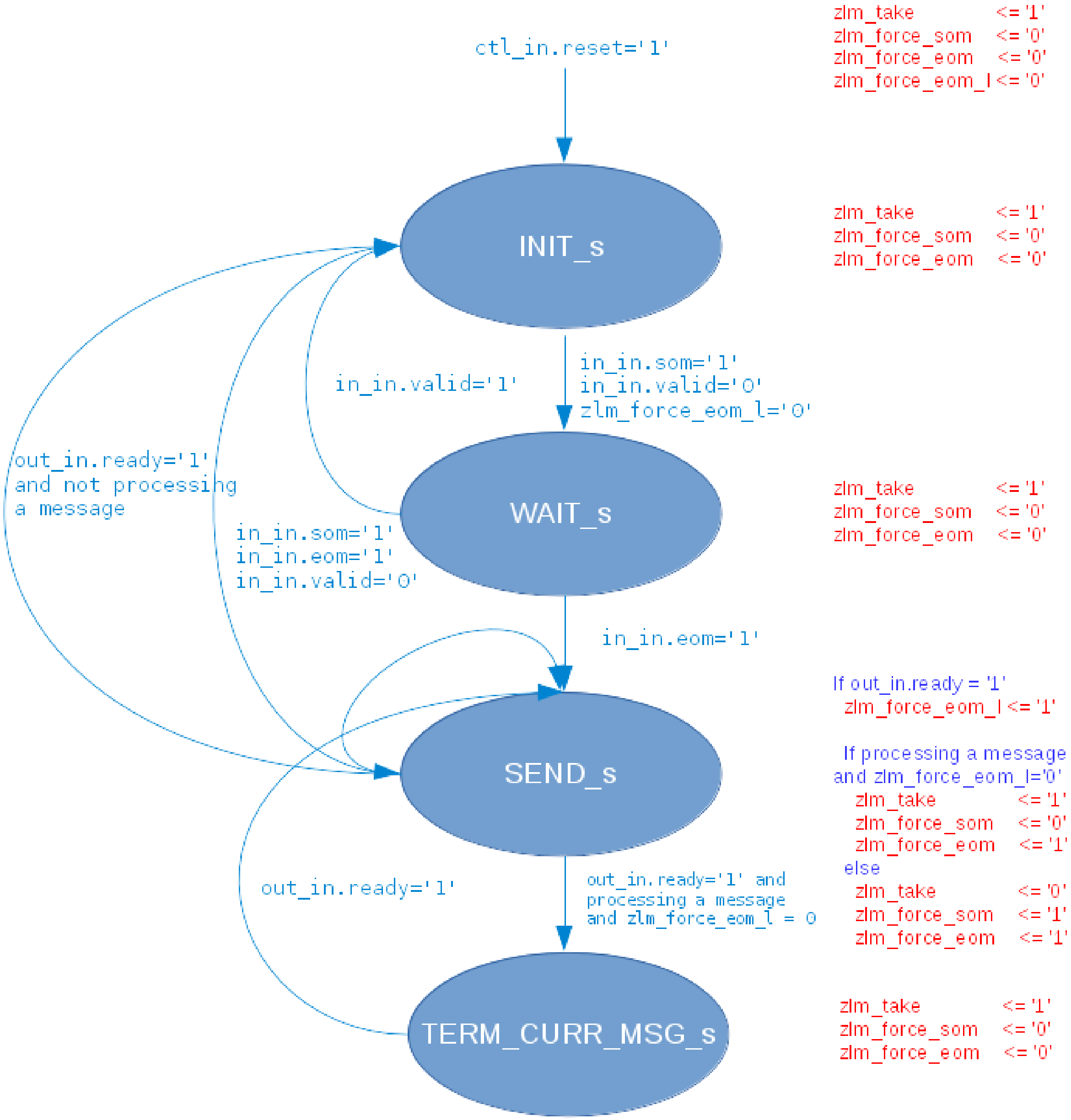

Figure 3: Zero-Length Message FSM

Note: In future releases this finite-state machine will be replaced with a register-delay based mechanism, currently exemplified in the dc offset filter

Source Dependencies

complex_mixer.rcc

- ocpi-prereq-liquid-1.2.0_*.rpm and ocpi-prereq-liquid-platform-zynq-1.2.0_*.rpm need to be installed in order to build and run this worker.

  /opt/opencpi/prerequisites/liquid/include/liquid/liquid.h

complex_mixer.hdl

- projects/assets/components/dsp_comps/complex_mixer.hdl/complex_mixer.vhd
- projects/assets/hdl/primitives/dsp_prims/dsp_prims_pkg.vhd

  projects/assets/hdl/primitives/dsp_prims/nco/src/nco.vhd

  projects/assets/hdl/primitives/dsp_prims/cordic/src/cordic_pr.vhd

  projects/assets/hdl/primitives/dsp_prims/cordic/src/cordic.vhd

  projects/assets/hdl/primitives/dsp_prims/cordic/src/cordic_stage.vhd
- projects/assets/hdl/primitives/util_prims/util_prims_pkg.vhd

  projects/assets/hdl/primitives/util_prims/mult/src/complex_mult.vhd

  projects/assets/hdl/primitives/util_prims/pd/src/peakDetect.vhd
- projects/assets/hdl/primitives/misc_prims/misc_prims_pkg.vhd

  projects/assets/hdl/primitives/misc_prims/round_conv/src/round_conv.vhd

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

Worker Properties

complex_mixer.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	`CHIPSCOPE_p`	Bool	-	-	Readable, Parameter	Standard	false	Include Chipscope circuit
Property	`NCO_DATA_WIDTH_p`	UChar	-	-	Readable, Parameter	12/16	12	Output data width of NCO
Property	`INPUT_DATA_WIDTH_p`	UChar	-	-	Readable, Parameter	12/16	12	Input port data width
Property	`CORDIC_STAGES_p`	UChar	-	-	Readable, Parameter	16	16	Number of CORDIC stages implemented in NCO
Property	`PEAK_MONITOR_p`	Bool	-	-	Readable, Parameter	Standard	true	Include peak monitor circuit
Property	`peak`	Short	-	-	Volatile	Standard	-	Output of peak detector
Property	`phs_init`	UShort	-	-	Readable, Writable	0	0	Initial phase of NCO

Property	`mag`	UShort	-	-	Readable, Writable	*	1024	Magnitude of NCO output, which must be in the range
								`[-2^(NCO_DATA_WIDTH_p-1)`
								`2^(NCO_DATA_WIDTH_p-1)-1]`
								in order for the worker to operate properly.

Property	`messageSize`	UShort	-	-	Readable, Writable	8192	8192	Number of bytes in output message
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	ZeroLengthMessages=true	Signed Complex Samples
Type	Name	DataWidth	Advanced	Usage
StreamInterface	out	32	ZeroLengthMessages=true	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 delay of `CORDIC_STAGES_p`+3 before the first sample is taken. After this initial delay, valid output data is given 2 clock cycles after input data is taken.

Latency
2 clock cycles

Worker Configuration Parameters

complex_mixer.hdl

Table 1: Table of Worker Configurations for worker: complex_mixer

Configuration	INPUT_DATA_WIDTH_p	ocpi_debug	VIVADO_ILA_p	CORDIC_STAGES_p	PEAK_MONITOR_p	NCO_DATA_WIDTH_p	ocpi_endian	CHIPSCOPE_p
0	12	false	false	16	true	12	little	false
1	16	false	false	16	true	16	little	false
2	12	false	true	16	true	12	little	false

Performance and Resource Utilization

Table 2: Resource Utilization Table for worker ”complex_mixer”

Configuration	OCPI Target	Tool	Version	Device	Registers (Typ)	LUTs (Typ)	Fmax (MHz) (Typ)	Memory/Special Functions
0	stratix4	Quartus	17.1.0	EP4SGX230KF40C2	1384	1804	N/A	DSP18: 8
0	zynq	Vivado	2017.1	xc7z020clg484-1	1328	2695	N/A	DSP48E1: 6
0	zynq_ise	ISE	14.7	7z020clg484-1	1325	2993	214.658	DSP48E1: 6
0	virtex6	ISE	14.7	6vlx240tff1156-1	1325	2996	222.218	DSP48E1: 6
1	stratix4	Quartus	17.1.0	EP4SGX230KF40C2	1584	1999	N/A	DSP18: 12
1	zynq	Vivado	2017.1	xc7z020clg484-1	1562	3219	N/A	DSP48E1: 6
1	zynq_ise	ISE	14.7	7z020clg484-1	1517	3491	162.76	DSP48E1: 8
1	virtex6	ISE	14.7	6vlx240tff1156-1	1517	3491	165.044	DSP48E1: 8
2	stratix4	Quartus	17.1.0	EP4SGX230KF40C2	1384	1804	N/A	DSP18: 8
2	zynq	Vivado	2017.1	xc7z020clg484-1	1329	2689	N/A	DSP48E1: 6
2	zynq_ise	ISE	14.7	7z020clg484-1	1326	2998	212.224	DSP48E1: 6
2	virtex6	ISE	14.7	6vlx240tff1156-1	1326	3001	222.218	DSP48E1: 6

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.