OpenCPI
IQ Imbalance Fixer Component Data Sheet

OpenCPI Release: v2.4.7

Revision History

Revision	Description of Change	Date
v1.4		10/2018
v1.5		4/2019
v1.6	Convert Worker to Version 2 HDL API	11/2019
v1.7	Table of Worker Configurations and Resource Utilization Table removed	5/2020

Summary - IQ Imbalance Fixer

Name	iq_imbalance_fixer
Worker Type	Application
OpenCPI Release	v2.4.7
Last Update	11/2019
Component Library	ocpi.assets.dsp_comps
Workers	iq_imbalance_fixer.hdl
Tested Platforms	alst4, E310(PL), isim, Matchstiq-Z1(PL), ml605, modelsim, xsim, ZedBoard(PL)

Functionality

The IQ Imbalance Fixer compensates for amplitude and phase differences between quadrature I and Q input rails caused by differences in quadrature A/D devices. The goal is to drive the corrected output power difference between rails to zero and to also drive the corrected phase difference between rails to zero. This is accomplished via a feedback loop where the power and phase differences are measured and applied to the Q rail. In other words, the I output is a delayed version of the I input while the Q output has been corrected by removing amplitude and phase errors. This results in removal of the input spectral image.

The power difference is measured by $I^2 - Q^2$, while the phase difference is measured by $I * Q$. These loop errors are averaged over $2^{+log2\_averaging\_length} - 1$ input samples, which defines the update interval. The loop errors are filtered by single-pole IIR filters at the end of the update interval. The filters have a pole on the unit circle and have gain of $2^{-neg\_log2\_loop\_gain}$. The filter outputs become the current *c_corr* and *d_corr* error correction values that are applied to the I and Q input streams, respectively. Input values are presented as corrected output values following four data valid cycles. This circuit operates at the full clock rate - that is, data valid may be held asserted every clock cycle.

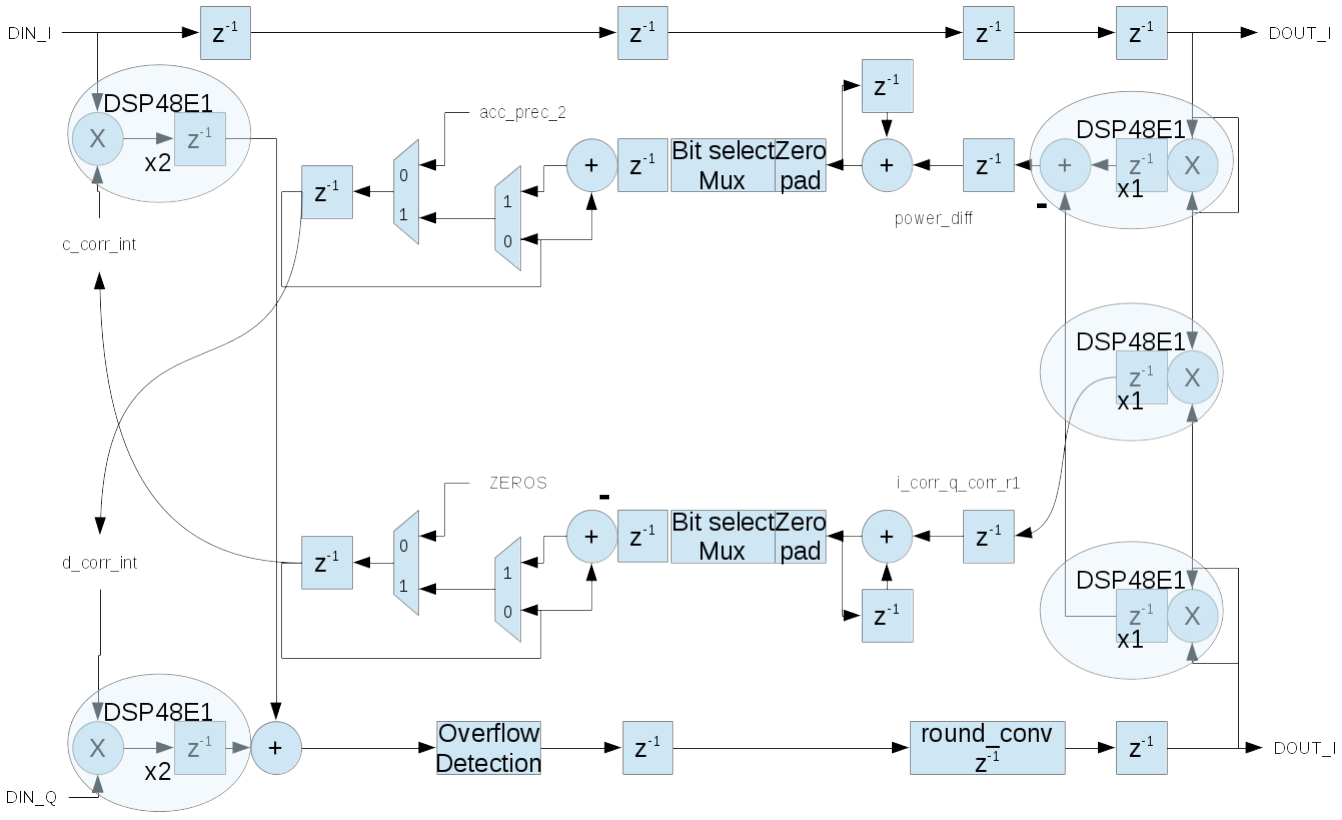


Figure 1: Block Diagram of VHDL, highlighting inferred Xilinx DSP48E1 primitives

Worker Implementation Details

iq_imbalance_fixer.hdl

The IQ Imbalance Fixer worker inputs complex signed samples and removes the spectral image caused by a quadrature gain imbalance between I and Q rails and also by a phase imbalance (not a perfect 90 degrees between sine and cosine) between the complex rails. The averaging time and the error loop gain of the worker are programmable, as is the ability to bypass the worker and to update/hold the calculated error correction values. For the HDL worker, a generic controls inclusion of a peak detection circuit applied to the worker's output samples.

An `ENABLE` input is available to either enable (true) or bypass (false) the circuit. Note that bypass registers are not used. Instead, feedback error values are held constant at values that do not alter the Q rail. The `UPDATE` input, by default true, may be disabled to hold the loop errors to a constant value. Note that the `ENABLE` input takes priority over the `UPDATE` input.

This implementation uses seven Xilinx DSP48e multipliers to process input data at the clock rate - i.e. this worker can handle a new input value every clock cycle. A peak detection circuit may be optionally included at build-time, which provides monitoring of the output amplitude, and may be used to influence the input gain. This worker will produce valid output four clock cycles after each valid input.

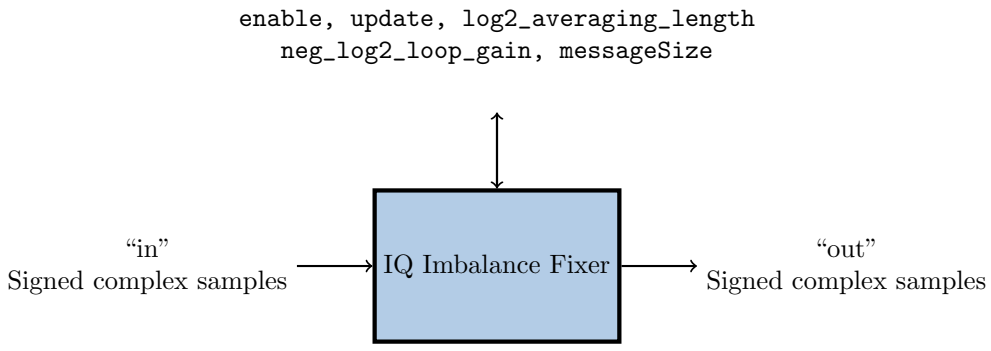
The IQ Imbalance Fixer worker utilizes the OCPI *iqstream_protocol* for both input and output ports. The *iqstream_protocol* defines an interface of 16-bit complex signed samples. The `DATA_WIDTH_p` parameter may be used to reduce the worker's internal data width to less than 16-bits.

Theory

A mathematical representation of a quadrature signal is given by the formula $cos(2\pi t/T) + jsin(2\pi t/T)$. If we let $\omega = 2\pi t/T$, the equation of an imbalanced quadrature signal is $cos(\omega) + j\alpha sin(\omega + \beta)$, where α is the amplitude difference and β is the phase difference between I/Q rails. If we take the imbalanced signal and mix it with another frequency, such as upconverting or downconverting the signal, we essentially multiply the input signal by the tone $cos(\omega 0) + jsin(\omega 0)$. The output signal then becomes a frequency-shifted version of the input signal given by $cos(\omega - \omega 0) + j\alpha sin(\omega - \omega 0 + \beta)$. The result is a spectral image of the original signal at $-(\omega - \omega 0)$. In the case of a non- zero linear combination of sine and cosine waves (which we have from a quadrature A/D, where the Q rail is simply a phase shift of $\pi/2$ of the I rail), we may use the trigonometric identity $\alpha sin(x + \beta) = Ccos(x) + Dsin(x)$, where C and D are constants and are the phase and amplitude errors applied to the I and Q input values, respectively. Thus the corrected output rails become I and $CI + DQ$.

Block Diagrams

Top level



Source Dependencies

iq_imbalance_fixer.hdl

- projects/assets/components/dsp_comps/iq_imbalance_fixer/iq_imbalance_fixer.vhd
- projects/assets/hdl/primitives/dsp_prims/dsp_prims_pkg.vhd
  projects/assets/hdl/primitives/dsp_prims/iq_imbalance/src/iq_imbalance_corrector.vhd
- projects/assets/hdl/primitives/util_prims/util_prims_pkg.vhd
  projects/assets/hdl/primitives/util_prims/pd/src/peakDetect.vhd

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
enable	Bool	-	-	Writable	Standard	true	Enable(true) or bypass(false)
update	Bool	-	-	Writable	Standard	true	Update the calculated amplitude and phase errors, or hold a previously calculated value
log2_averaging_length	UChar	-	-	Writable	1-31	11	Controls the update interval to be applied to the input, where $2^n + 1$ samples define the averaging length
neg_log2_loop_gain	UChar	-	-	Writable	1-31	5	Controls the loop gain, where the value is 2^{-n}
messageSize	UShort	-	-	Writable	8192	8192	Number of bytes in output message (Not implemented by Version 2)

Worker Properties

iq_imbalance_fixer.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	DATA_WIDTH_p	UChar	-	-	Parameter	1-16	16	Data Width of I and Q for internal processing
Property	ACC_PREC_p	UChar	-	-	Parameter	3-?	34	Number of bits of precision for accumulator
Property	PEAK_MONITOR_p	Bool	-	-	Parameter	Standard	true	Include a peak detection circuit
Property	peak	Short	-	-	Volatile	Standard	0	Peak value of I or Q output (valid when PEAK_MONITOR_p=true)

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

iq_imbalance_fixer.hdl

Type	Name	DataWidth	Advanced	Usage
StreamInterface	in	32		Signed complex samples
StreamInterface	out	32	InsertEOM=1	Signed complex samples

Control Timing and Signals

The IQ Imbalance Fixer worker uses the clock from the Control Plane and standard Control Plane signals.

Worker Configuration Parameters

iq_imbalance_fixer.hdl

Performance and Resource Utilization

iq_imbalance_fixer.hdl

Test and Verification

A single test case is implemented to validate the IQ Imbalance Fixer component. An input file is generated consisting of a waveform with tones at 5 Hz, 13 Hz, and 27 Hz, and creates a spectral image by both adding different gain to I and Q rails and also adding an additional 10 degrees of phase to the sine portion of the complex signal. The result is the addition of spectral image tones at -5 Hz, -13 Hz, and -27 Hz. The complex waveform is then scaled to fixed-point signed 16-bit integers, with a maximum amplitude of 31000 to avoid roll-over.

Time and frequency domain plots may be viewed in Figures 2 and 3 below, respectively, where the time domain plot represents the first 128 complex samples in the file.

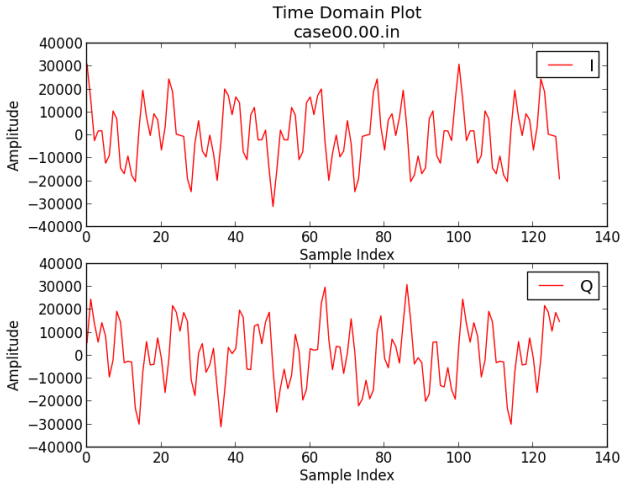


Figure 2: Time Domain Tones with Spectral Image

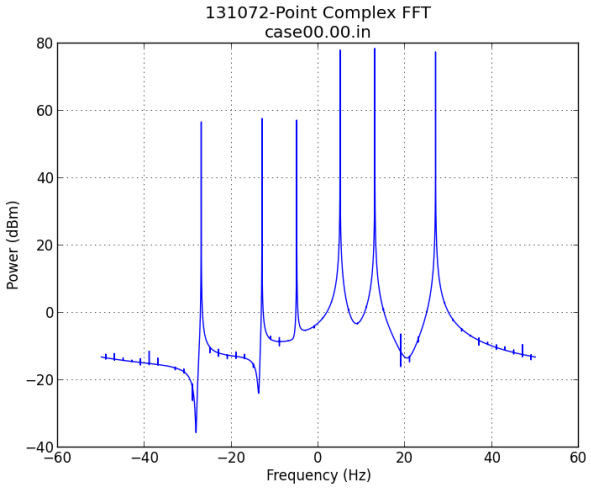


Figure 3: Frequency Domain Tones with Spectral Image

Verification of output data is performed only on the second half of the output, which represents the steady state of the component. The output is first checked that the data is not all zero, and is then checked for the expected length of 65,536 complex samples. Once these quick checks are made both the input and output data are translated to the frequency domain, where a FFT is performed, and then power measurements are taken at 5 Hz, 13 Hz, 27 Hz, -5 Hz, -13 Hz, and -27 Hz. The input and output power measurements are compared to validate that the IQ spectral image has been attenuated and that the other tones have not been attenuated. In addition, the FFT bin with the maximum power in the range of DC to +Fs/2 is compared to the maximum power bin in the range of -Fs/2 to DC to verify at least 65 dB of suppression has taken place with respect to the spectral image. Figures 4 and 5 depict the filtered results of the three tone input, where the time domain plot represents the first 128 complex samples in the file.

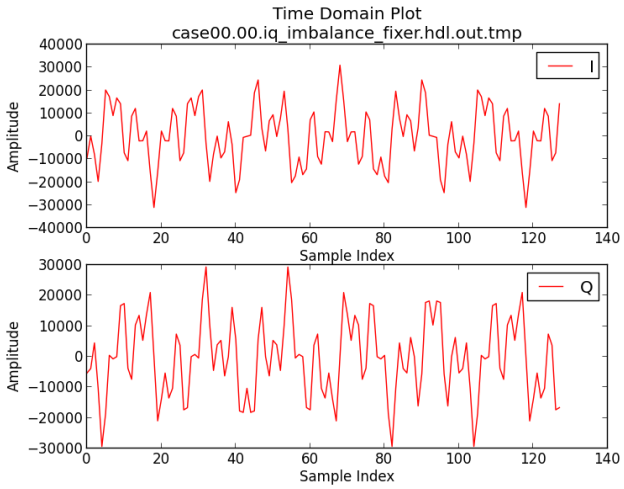


Figure 4: Time Domain Tones with Spectral Image re-moved

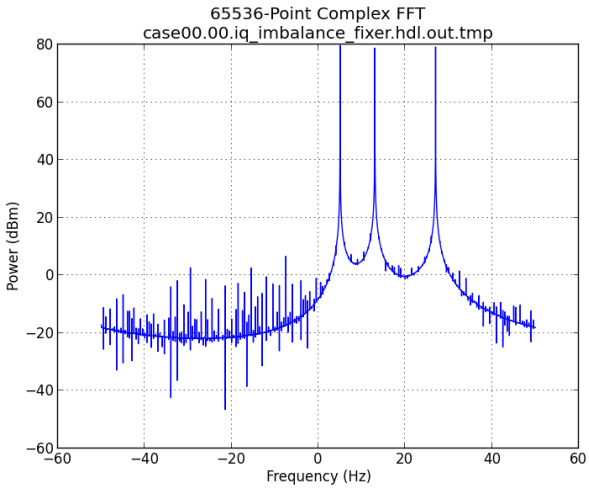


Figure 5: Frequency Domain Tones with Spectral Image removed