

OpenCPI

IQstream To CSWM Component Data Sheet

OpenCPI Release: v2.4.7

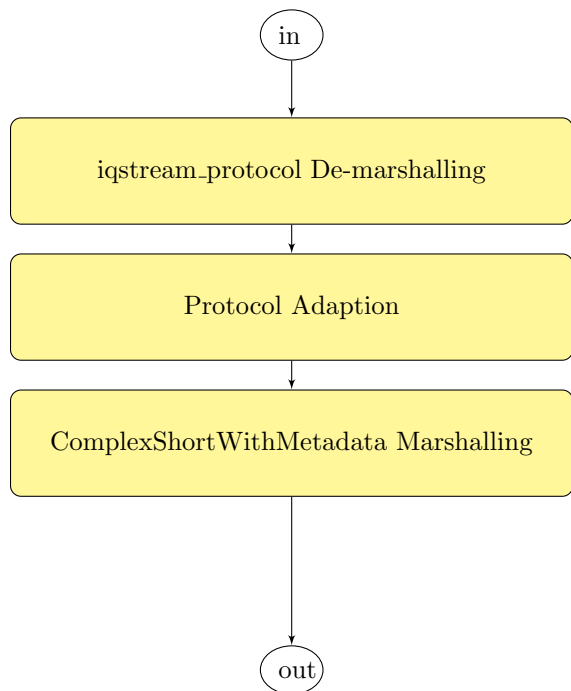
Revision History

Revision	Description of Change	Date
v1.7	Initial release	05/2020

Summary - IQstream To CSWM

Name	iqstream_to_cswm
Worker Type	Application
OpenCPI Release	v2.4.7
Last Update	05/2020
Component Library	ocpi.assets.misc_comps
Workers	iqstream_to_cswm.hdl
Tested Platforms	CentOS7, xsim

Block Diagram



Functionality

"It then converts them to ComplexShortWithMetadata protocol messages that are marshaled on to the out port, according to the following table."

iqstream operation	ComplexShortWithMetadata operation
iq	samples
	time
	interval
	flush
	sync
	end_of_samples

Source Dependencies

cswm_to_iqstream.hdl

- core/hdl/primitives/protocol/complex_short_with_metadata_body.vhd
- core/hdl/primitives/protocol/complex_short_with_metadata_marshall.vhd

- core/hdl/primitives/protocol/complex_short_with_metadata_pkg.vhd
- core/hdl/primitives/protocol/iqstream_demarshaller.vhd
- core/hdl/primitives/protocol/iqstream_pkg.vhd
- core/hdl/primitives/protocol/protocol_pkg.vhd

Test and Verification

The Unit Under Test (UUT) is tested by a) generating a file containing IQStream protocol-formatted messages which are sent to the UUT via the `file_read` component, b) using the `file_write` component to save the file containing the UUT's output as protocol-formatted message, and c) verifying the contents of the output file. The generated file contains one or more `ComplexShortWithMetadata samples` messages. The sequence of `samples` messages include samples whose I and Q signals are positive and negative ramp functions, respectively. The output file is verified by confirming that a) the same number of samples exists in the input and output files, and b) the output file contains the same ramp functions as the input.

Output Width and Bit Growth

Both the input and output data is marshalled across a 32-bit bus, so bit growth is not a concern.