OpenCPI
Phase to Amplitude CORDIC Component Data Sheet

OpenCPI Release: v2.3.4

Revision History

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

Summary - Phase to Amplitude CORDIC

Name	phase_to_amp_cordic
Worker Type	Application
OpenCPI Release	v2.3.4
Last Update	11/2019
Component Library	ocpi.assets.dsp_comps
Workers	phase_to_amp_cordic.hdl, phase_to_amp_cordic.rcc
Tested Platforms	alst4, CentOS7, E310(PL), isim, Matchstiq-Z1(PL), ml605, modelsim, xilinx13_3, xsim, ZedBoard(PL)

Functionality

This worker implements a phase to amplitude conversion (PAC). The real 16 bit signed input data is phase accumulated, then fed into a polar-to-rectangular CORDIC. The output of the CORDIC produces a complex waveform. Figure 1 diagrams the Phase to Amplitude CORDIC.

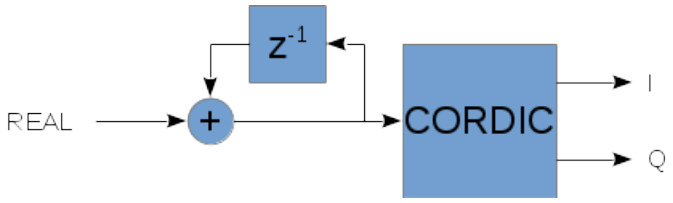

Figure 1: Phase to Amplitude CORDIC Functional Diagram

Worker Implementation Details

phase_to_amp_cordic.hdl

The phase to amplitude converter calculates the amplitude for the current phase angle. This operation is basically the same as calculating the sine or cosine function of its argument. Two methods are typically used for implementing in hardware, the Coordinate Rotation Digital Computer (CORDIC) algorithm and the ROM lookup table.

This worker implements the CORDIC algorithm. The frequency of its complex output is determined by the following equation:

$$output\_freq = \frac{phs\_accum}{2^{DATA\_WIDTH}} \quad (1)$$

Where `phs_accum` is the output of the accumulator. `DATA_WIDTH` is the input/output data width of the CORDIC which has a range of 8 to 16. The input clock frequency is the sample rate of the samples. The amplitude of the complex wave is runtime configurable via the `magnitude` property. An `enable` input is available to either enable (true) or bypass (false) the circuit. In bypass mode, pipe-lining registers are not used, and the real input data is available on the lower 16 bits of the complex output.

Build time parameters control the width of the input, output and the number of stages of the CORDIC primitive module. The I/O data widths of the worker itself are set within the OCS and adjusted in the OWD.

phase_to_amp_cordic.rcc

This RCC worker is intended to be used in an all-RCC worker version of the FSK application, as it is currently implemented. Where components are incorrectly named to reflect their implementation, rather than based on their functionality, i.e. phase_to_amp_cordic vs polar_rect.

Additionally, this worker implements a secondary function (i.e. phase accumulator), which does not respect the component (1 function) model. For these reasons, and those discussed below, it is highly recommended that the usage of this RCC worker be limited to the all-RCC FSK applications and not used for new applications.

This RCC (C++) worker is a work-a-like to the HDL worker, similarly named phase_to_amp_cordic.hdl. However, it does NOT implement a CORDIC algorithm (i.e. stages of shifts & adds), but rather utilizes floating point cmath functions to calculate its output. Due to HDL worker implementation decisions, which affect performance, the RCC worker was required to implement additional (restrictive) functionality to emulate the HDL worker's behavior.

The calculations that it does first is phase accumulating the incoming real samples. Then iterates over the phase values and converts from phase to radians as the input to the cmath sin and cos functions. The resulting value is multiplied by the magnitude to produce the scaled output given to the output port.

The HDL worker will not output the final samples depending on the number of CORDIC stages and pipeline stages. To match the HDL implementation, the rcc worker implements a stage delay using a deque as a FIFO (in_FIFO). The FIFO contains the previous input samples that would still be present in the HDL implementation's pipline. Calculations are made when data is present in the FIFO and enough data on the input port is present to push more data out. Then on the input data phase calculations are made.

Limitations: This worker does not implement the CORDIC (shifts-add) algorithm, but equivalent floating point math with fixed-point adjustments (using float to integer truncation)

Block Diagrams

Top level

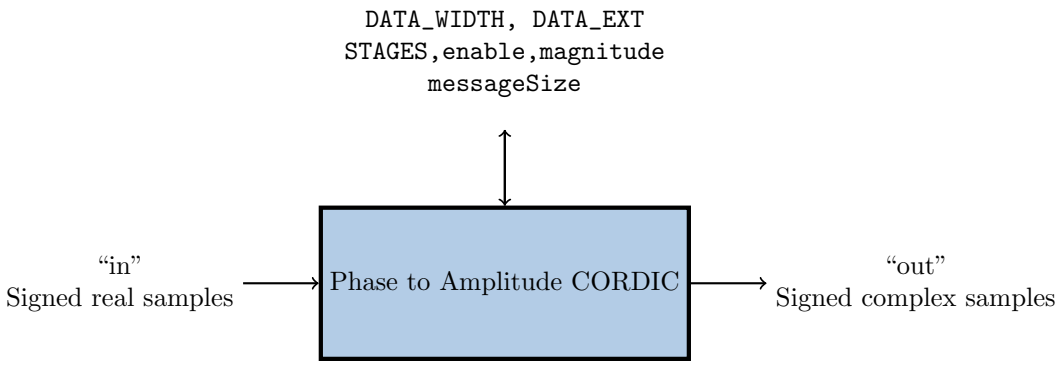

Figure 2: Top Level Block Diagram

Source Dependencies

phase_to_amp_cordic.hdl

- projects/assets/components/dsp_comps/phase_to_amp_cordic.hdl/phase_to_amp_cordic.vhd
- projects/assets/hdl/primitives/dsp_prims/dsp_prims_pkg.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/misc_prims/misc_prims_pkg.vhd
  projects/assets/hdl/primitives/misc_prims/round_conv/src/round_conv.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
DATA_WIDTH	UChar	-	-		-	-	Input (real) and Output (I/Q) data width
DATA_EXT	UChar	-	-		-	-	CORDIC requirement: # of extension bits
STAGES	UChar	-	-		-	-	Number of CORDIC stages implemented
messageSize	UShort	-	-	Writable	-	8192	Number of bytes in output message (Not implemented by Version 2)
enable	Bool	-	-	Writable	-	True	Enable(true) or bypass(false)
magnitude	UShort	-	-	Writable	-	16384	Magnitude of output * +2^(DATA_WIDTH)-1 to -2^(DATA_WIDTH)

Worker Properties

phase_to_amp_cordic.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
SpecProperty	DATA_WIDTH	-	-	-	Parameter	8-16	16	Input (real) and Output (I/Q) data width
SpecProperty	DATA_EXT	-	-	-	Parameter	6	6	CORDIC requirement: Number of extension bits
SpecProperty	STAGES	-	-	-	Parameter	8-16	12	Number of CORDIC stages implemented
Property	PEAK_MONITOR	Bool	-	-	Parameter	Standard	true	Enable/Disable build-time inclusion of peak monitor circuit
Property	peak	Short	-	-	Volatile	Standard	0	Peak value of I/Q output (valid when PEAK_MONITOR=true)

phase_to_amp_cordic.rcc

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
SpecProperty	DATA_WIDTH	-	-	-	Parameter	8-16	16	Input (real) and Output (I/Q) data width
SpecProperty	DATA_EXT	-	-	-	Parameter	6	6	CORDIC requirement: Number of extension bits
SpecProperty	STAGES	-	-	-	Parameter	8-16	12	Number of CORDIC stages implemented
Property	PEAK_MONITOR	Bool	-	-	Parameter	Standard	true	Enable/Disable build-time inclusion of peak monitor circuit
Property	peak	Short	-	-	Volatile	Standard	0	Peak value of I/Q output (valid when PEAK_MONITOR=true)
Property	AdditionalDelay	UChar	-	-	Parameter	Standard	2	Additional number of delays over CORDIC stages (STAGES) to match HDL implementation.
Property	StageDelay	UChar	-	-	Parameter	0-255	STAGES + AdditionalDelay	Number of delays to match HDL implementation.

Component Ports

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

Worker Interfaces

phase_to_amp_cordic.hdl

Type	Name	DataWidth	Advanced	Usage
StreamInterface	in	16		Signed real samples
StreamInterface	out	32	InsertEOM=1	Signed complex samples

Control Timing and Signals

The Phase to Amplitude CORDIC worker uses the clock from the Control Plane and standard Control Plane signals. This worker has a start-up transient delay of `STAGES`+2 valid samples. This means that once the input is ready and valid and the output is ready, it takes that many valid input samples before the first output sample is made available. The delays for this worker are itemized be as follows:

- 0 : pr_cordic.vhd
- 1 : pr_cordic.vhd/cordic_pr.vhd
- STAGES : pr_cordic.vhd/cordic_pr.vhd/cordic.vhd/cordic_stage.vhd
- 1 : cordic_pr.vhd/round_conv.vhd

Latency
`STAGES`+2 clock cycles

Worker Configuration Parameters

phase_to_amp_cordic.hdl

Performance and Resource Utilization

phase_to_amp_cordic.hdl

Test and Verification

This component is tested via the unit test automation feature of the framework. The component's .test/ contains XML files that describe the combinations of tests.

Two test cases are employed to verify the Phase to Amplitude CORDIC component:

1. Disabled. The real input data is passed through the worker and made available on lower 16 bits of the complex output.

2. Constant output frequency: A python script creates a file containing a constant value. The CORDIC produces a complex output frequency according to the equation 1.

The plots below show the input (real) and output data (Q-leg showing all samples, I-leg zoomed into one cycle) for testing a Phase to Amplitude CORDIC having the default parameter set. The input file shows a DC value of 2048, and 16384 real samples. The output file shows the converted complex waveform having $DCvalue/2^{DATA\_WIDTH}$=$2048/2^{16}$=0.03125 Hz, and 16384 complex samples.

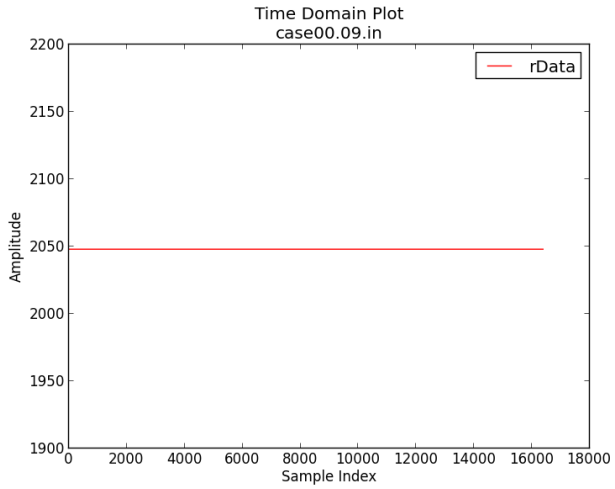

Figure 3: Time Domain

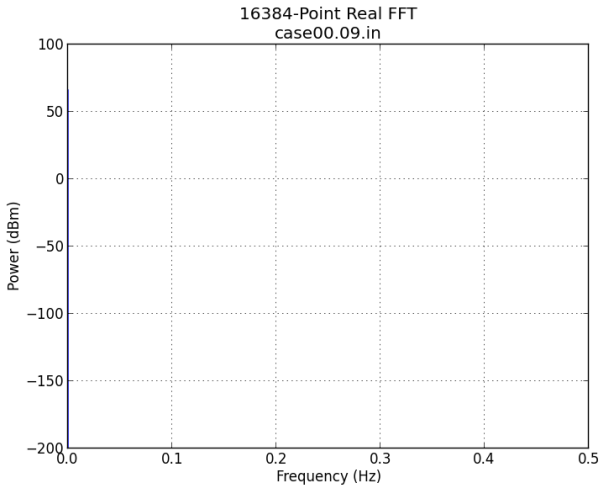

Figure 4: Frequency Domain

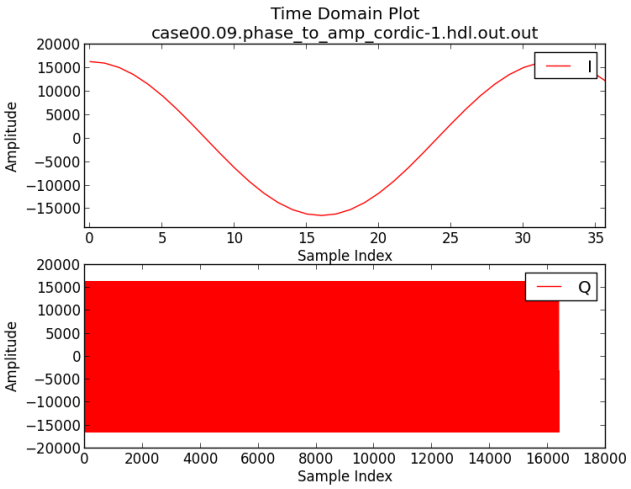

Figure 5: Time Domain

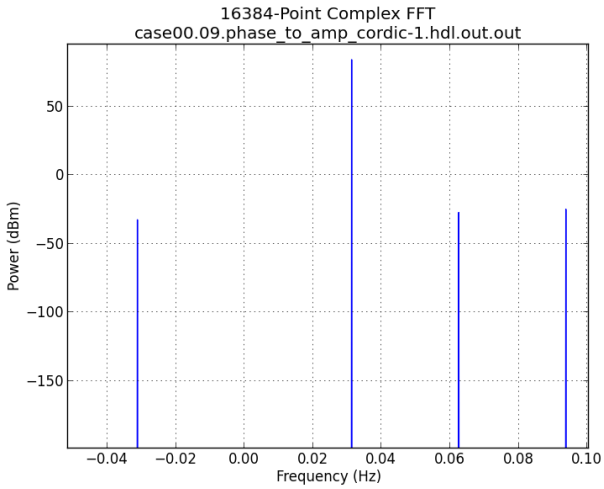

Figure 6: Frequency Domain