Summary - Zero Pad

Name	zero_pad
Worker Type	Application
Version	v1.5
Release Date	4/2019
Component Library	ocpi.assets.util_comps
Workers	zero_pad.hdl
Tested Platforms	xsim, isim, modelsim, Matchstiq-Z1(PL)

Functionality

The Zero Pad component inputs samples and inserts a configurable number of zeros between output samples.

Worker Implementation Details

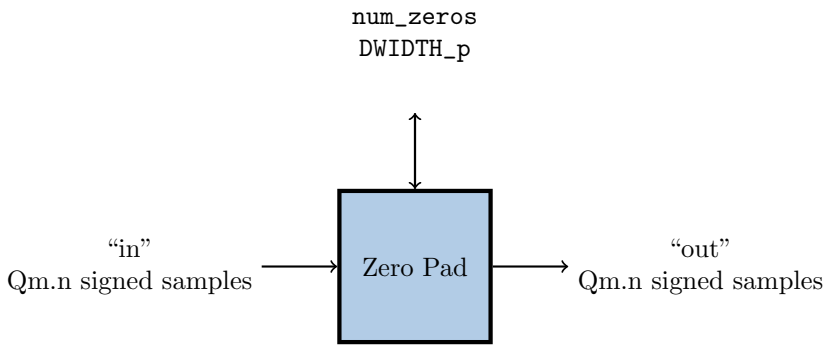
In order to be maximally flexible, the component does not define input/output protocols explicitly. The input/output data widths are defined at build time, which in turn define the respective input/output sample sizes.

The input and output data width are the same and are configurable via the `DWIDTH_p` parameter. All zeros inserted between samples are also of size `DWIDTH_p`.

The message size on the output is equal to the number of zeros inserted plus 1.

Block Diagrams

Top level



Source Dependencies

zero_pad.hdl

- projects/assets/components/util_comps/zero_pad.hdl/zero_pad.vhd

zero_pad.rcc

- projects/assets/components/util_comps/zero_pad.rcc/zero_pad.cc

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
DWIDTH_p	ULong	-	-	Readable, Parameter	8,16,32,64	16	Input and output port data width
num_zeros	Short	-	M_p	Readable, Writable	Standard	-	Number of zeros to be inserted between output samples

Worker Properties

There are no worker implementation-specific properties for this component

Component Ports

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

Worker Interfaces

zero_pad.hdl

Type	Name	DataWidth	Advanced	Usage
StreamInterface	in	DWIDTH_p	-	Signed real samples
StreamInterface	out	DWIDTH_p	-	Signed real samples

Control Timing and Signals

The Zero Pad HDL worker uses the clock from the Control Plane and standard Control Plane signals.

Worker Configuration Parameters

zero_pad.hdl

Table 1: Table of Worker Configurations for worker: zero_pad

Configuration	ocpi_debug	DWIDTH_p	ocpi_endian
0	false	8	little
1	false	16	little
2	false	32	little
3	false	64	little

Performance and Resource Utilization

zero_pad.hdl

Table 2: Resource Utilization Table for worker "zero_pad"

Configuration	OCPI Target	Tool	Version	Device	Registers (Typ)	LUTs (Typ)	Fmax (MHz) (Typ)	Memory/Special Functions
0	stratix4	Quartus	17.1.0	EP4SGX230KF40C2	225	262	N/A	N/A
0	zynq	Vivado	2017.1	xc7z020clg484-1	223	289	N/A	N/A
0	zynq_ise	ISE	14.7	7z020clg484-1	221	368	199.601	N/A
0	virtex6	ISE	14.7	6vlx240tff1156-1	221	374	189.251	N/A
1	stratix4	Quartus	17.1.0	EP4SGX230KF40C2	260	282	N/A	N/A
1	zynq	Vivado	2017.1	xc7z020clg484-1	258	310	N/A	N/A
1	zynq_ise	ISE	14.7	7z020clg484-1	256	401	196.58	N/A
1	virtex6	ISE	14.7	6vlx240tff1156-1	257	432	186.916	N/A
2	stratix4	Quartus	17.1.0	EP4SGX230KF40C2	331	298	N/A	N/A
2	zynq	Vivado	2017.1	xc7z020clg484-1	329	377	N/A	N/A
2	zynq_ise	ISE	14.7	7z020clg484-1	327	458	195.465	N/A
2	virtex6	ISE	14.7	6vlx240tff1156-1	330	512	187.617	N/A
3	stratix4	Quartus	17.1.0	EP4SGX230KF40C2	474	319	N/A	N/A
3	zynq	Vivado	2017.1	xc7z020clg484-1	472	474	N/A	N/A
3	zynq_ise	ISE	14.7	7z020clg484-1	468	594	219.78	N/A
3	virtex6	ISE	14.7	6vlx240tff1156-1	468	616	205.339	N/A

Test and Verification

Data widths of 8/16/32/64 are supported and fully tested on both RCC and HDL worker implementations. These widths combinations are tested with `num_zeros` equal to 0, 7, 8, 38, 127, and 255.

Input data is generated by a python script with an input parameter that defines the number of 32-bit words to produce. The input file consists of a repeating pattern of 0x0123456789ABCDEF. The number of 32-bit words for each test case is 2048, which results in 1024 64-bit samples, 2048 32-bit samples, 4096 16-bit samples, or 8192 8-bit samples. Thus for each test case the 64-bit test pattern is repeated 1024 times to produce a file of 65,536 bits (or 8192 bytes).

The Zero Pad component inputs each sample and `num_zeros` zeros are inserted between each output sample.

For verification, the output file is first checked that the data is not all zero, and is then checked for the expected length. Once these quick checks are made the output data is compared against expected results sample-by-sample without use of any gold files.