OpenCPI
FIR Complex SSE Component Data Sheet

OpenCPI Release: v2.4.6

Revision History

Revision	Description of Change	Date
v1.4		10/2018
v1.5		4/2019
v1.6	Converted Worker to version 2	11/2019

Summary - FIR Complex SSE

Name	fir_complex_sse
Worker Type	Application
OpenCPI Release	v2.4.6
Last Update	11/2019
Component Library	ocpi.assets.dsp_comps
Workers	fir_complex_sse.hdl
Tested Platforms	alst4, isim, Matchstiq-Z1(PL), ml605, modelsim, xsim, ZedBoard(PL)

Functionality

The FIR Complex SSE (Systolic Symmetric Even) component inputs complex signed samples and filters them based upon a programmable number of coefficient tap values. The underlying FIR Filter implementation makes use of a symmetric systolic structure to construct a filter with an even number of taps and symmetry about its midpoint.

Worker Implementation Details

fir_complex_sse.hdl

Filter taps

The `NUM_TAPS_p` parameter determines the length of the `taps` property. Each 16-bit tap within the `taps` property is truncated by taking the `COEFF_WIDTH_p` number of least significant bits. Care should be taken to ensure that the `COEFF_WIDTH_p` parameter is ≤ 16. The most significant bit of the truncated value is used as a sign bit. The truncated data is formatted as QN.0, where N is `COEFF_WIDTH_p`-1. The truncated taps are effectively the first half of all coefficients used to perform convolution. The last half is equivalent to the reverse-indexed copy of the first half, allowing the coefficients to have symmetry about their midpoint. This symmetry of FIR coefficients results in group delay being constant across all frequencies.

Filter function

This implementation uses `NUM_TAPS_p` multipliers per each input rail to process input data at the clock rate - i.e. this worker can handle a new input value every clock cycle. It is unnecessary to round the output data from this filter at the worker level because it is already being done within the macc_systolic_sym primitive.

The FIR Complex SSE 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 FIR Complex SSE hdl worker's internal data width to less than 16-bits.

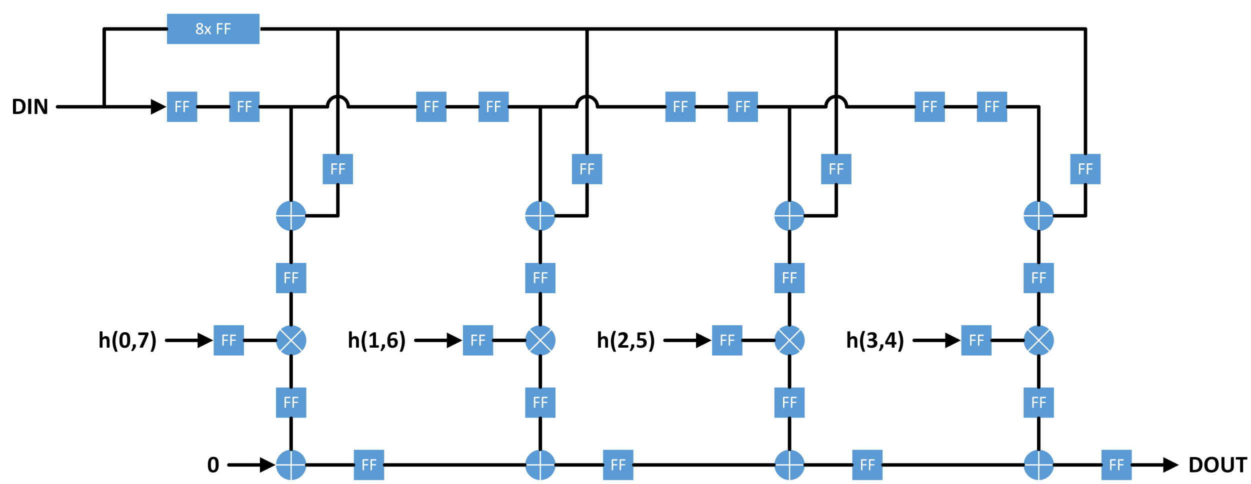


Figure 1: FIR Complex SSE Block Diagram - 8-tap example per I/Q rail

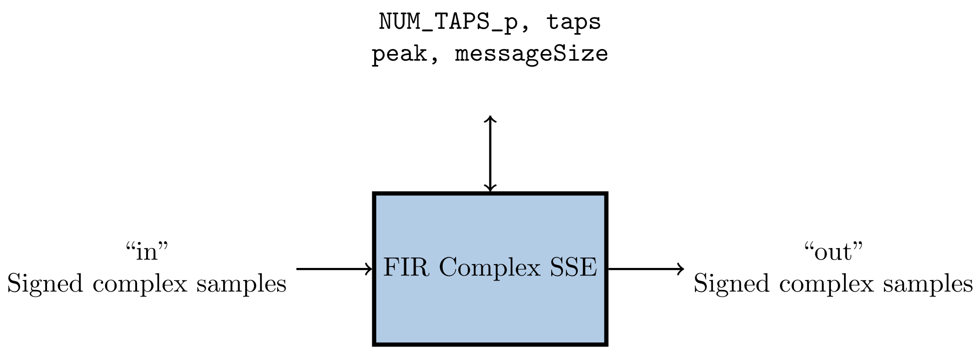
Theory

This filter will produce valid outputs one clock after each valid input, but care must be exercised when attempting to align outputs according to the filter's actual group delay and propagation delay.

For a FIR filter with symmetric impulse response we are guaranteed to have linear phase response and thus constant group delay vs. frequency. In general, the group delay will be equal to (N-1)/2, where N is the number of filter taps. The filter topology itself will add some propagation delay to the response. For this design the total delay from an impulse input to the beginning of the impulse response will be `NUM_TAPS_p` + 4 samples.

Block Diagrams

Top level



Source Dependencies

fir_complex_sse.hdl

- projects/assets/components/dsp_comps/fir_complex_sse.hdl/fir_complex_sse.vhd
- projects/assets/hdl/primitives/dsp_prims/dsp_prims_pkg.vhd
  projects/assets/hdl/primitives/dsp_prims/fir/src/fir_systolic_sym_even.vhd
  projects/assets/hdl/primitives/dsp_prims/fir/src/macc_systolic_sym.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
NUM_TAPS_p	UChar	-	-	Parameter	1-?	16	Half the number of coefficients used by each real/imag even symmetric filter
peak	Short	-	-	Volatile	Standard	0	Peak value of I/Q output (valid when PEAK_MONITOR = true)
messageSize	UShort	-	-	Writable	8192	8192	Number of bytes in output message (Not implemented by Version 2)
taps	Short	-	NUM_TAPS_p	Initial	$-2^{\mathrm{COEFF\_WIDTH\_p}-1}$ to $+2^{\mathrm{COEFF\_WIDTH\_p}-1}-1$	-	Symmetric filter coefficient values loaded into both real/imag filters

Worker Properties

fir_complex_sse.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	COEFF_WIDTH_p	UChar	-	-	Parameter	1-32	16	Coefficient width in bits.
Property	PEAK_MONITOR	bool	-	-	Parameter	-	true	Enable/Disable build-time inclusion of Peak Monitoring

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
in	false	iqstream_protocol	false	-	Complex signed samples
out	true	iqstream_protocol	false	-	Complex signed samples

Worker Interfaces

fir_complex_sse.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 FIR Complex SSE worker uses the clock from the Control Plane and standard Control Plane signals. The Raw Property interface is used to read/write coefficient values.

Worker Configuration Parameters

fir_complex_sse.hdl

Performance and Resource Utilization

fir_complex_sse.hdl

Test and Verification

A single test case is implemented to validate the FIR Complex SSE component. The python script *gen_lpf_taps.py* is used to generate a taps file consisting of `NUM_TAPS_p` filter coefficients. Input data is generated by first creating a *.dat input file consisting of a single maximum signed value of +32767 (for each real/imag filter) followed by 2*(`NUM_TAPS_p`-1) zero samples (again for each real/imag filter). The *.bin input file is the binary version of the *.dat ASCII file repeated 2*`NUM_TAPS_p` times.

The FIR Complex SSE worker inputs complex signed samples, filters the input as defined by the coefficient filter taps, and outputs complex signed samples. Since the input consists of an impulse response - that is, a maximal 'one' sample followed by all zeros equal to the length of the filter - the output of each filter is simply the coefficient values.

For verification, the output file is first checked that both I and Q outputs match. Then the I output rail is compared against the taps file, and finally the Q output rail is also compared against the taps file. A ±1 difference is allowed in value while comparing output rails against the filter coefficient values. Figures 2 and 3 depict the filtered results of the impulse input.

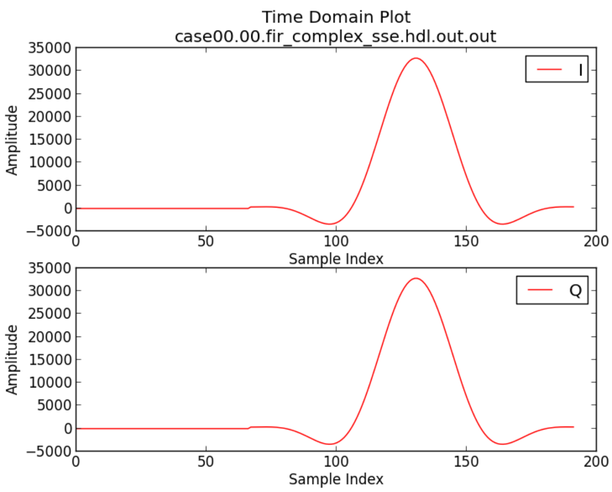


Figure 2: Time Domain Impulse Response

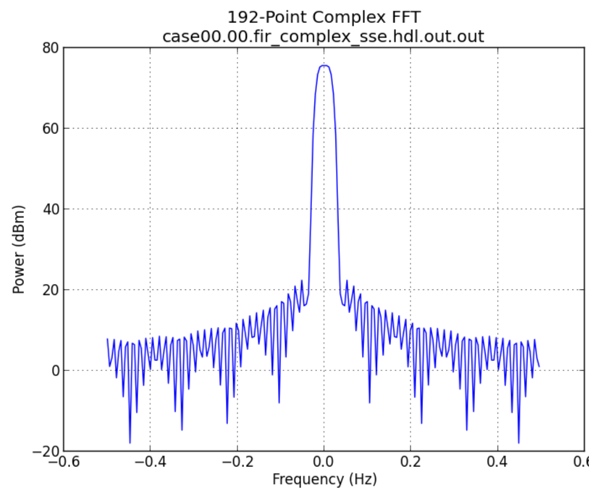


Figure 3: Frequency Domain Impulse Response