

## Summary - FIR Complex SSE

Name	fir_complex_sse
Worker Type	Application
Version	v1.3
Release Date	February 2018
Component Library	ocpi.assets.dsp_comps
Workers	fir_complex_sse.hdl
Tested Platforms	xsim, isim, modelsim, alst4, ml605, ZedBoard(PL), Matchstiq-Z1(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

The NUM\_TAPS\_p parameter defines  $N/2$  coefficient values. Care should be taken to ensure that the COEFF\_WIDTH\_p parameter is  $\leq$  the type (size) of the taps property - i.e. a COEFF\_WIDTH\_p of 1-8 should use a taps type of char; a COEFF\_WIDTH\_p of 9-16 should use a taps type of short; and a COEFF\_WIDTH\_p of 17-32 should use a taps type of long. Identical filter tap coefficients are applied to both real and imaginary input samples.

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 worker's internal data width to less than 16-bits.

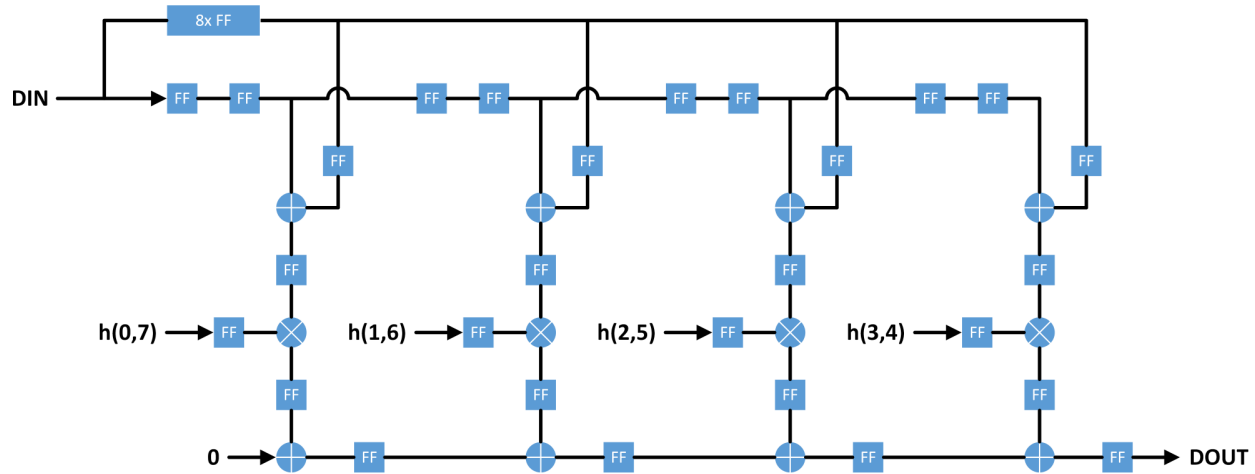


Figure 1: FIR Complex SSE Block Diagram - 8-tap example per I/Q rail

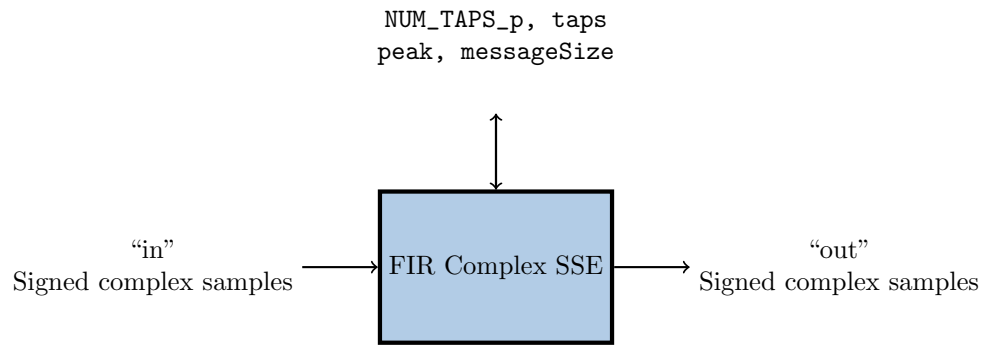
## Theory

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  $\text{NUM\_TAPS\_p} + 4$  samples.

## Block Diagrams

### Top level



### State Machine

Only one finite-state machine (FSM) is implemented by this worker. The FSM supports Zero-Length Messages.

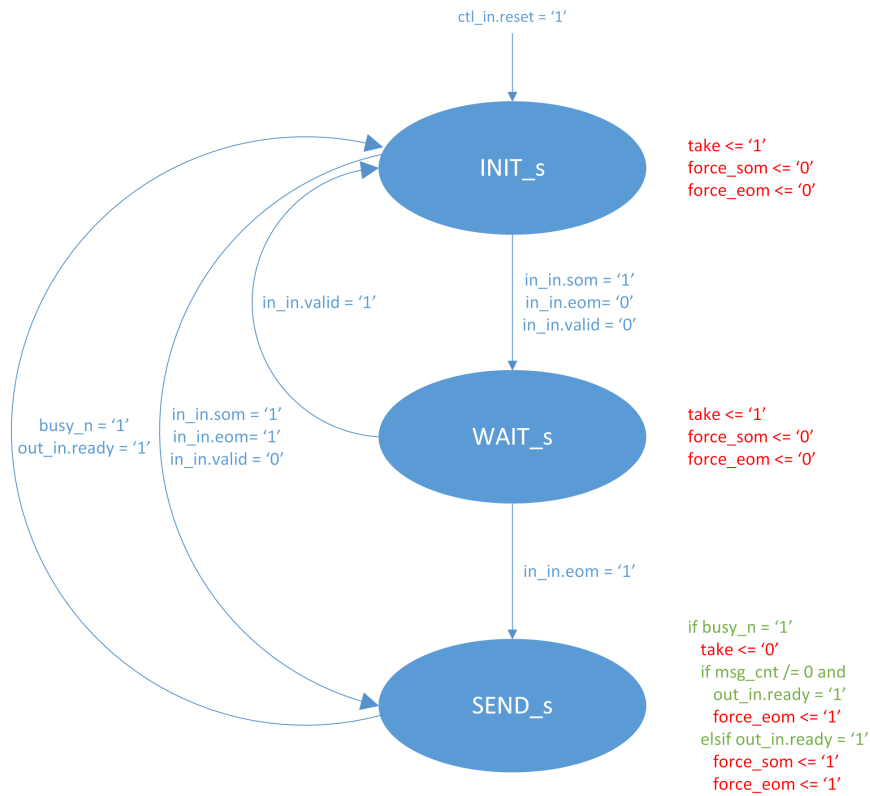


Figure 2: Zero-Length Message FSM

## Source Dependencies

### **fir\_complex\_sse.hdl**

- ocpiassets/components/dsp\_comps/fir\_complex\_sse.hdl/fir\_complex\_sse.vhd
- ocpiassets/hdl/primitives/dsp\_prims/dsp\_prims\_pkg.vhd
  - ocpiassets/hdl/primitives/dsp\_prims/fir/src/fir\_systolic\_sym\_even.vhd
  - ocpiassets/hdl/primitives/dsp\_prims/fir/src/macc\_systolic\_sym.vhd
- ocpiassets/hdl/primitives/misc\_prims/misc\_prims\_pkg.vhd
  - ocpiassets/hdl/primitives/misc\_prims/round\_conv/src/round\_conv.vhd
- ocpiassets/hdl/primitives/util\_prims/util\_prims\_pkg.vhd
  - ocpiassets/hdl/primitives/util\_prims/pd/src/peakDetect.vhd

## Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
NUM_TAPS_p	ULong	-	-	Readable, Parameter	1-?	16	Half the number of coefficients used by each real/imag even symmetric filter
peak	Short	-	-	Volatile	Standard	0	Read-only amplitude which may be useful for gain control
messageSize	UShort	-	-	Writable, Readable	8192	8192	Number of bytes in output message
taps	Short	-	NUM_TAPS_p	Writable, Readable	$-2^{\text{COEFF\_WIDTH\_p}-1}$ to $+2^{\text{COEFF\_WIDTH\_p}-1}-1$	-	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	ULong	-	-	Readable, Parameter	1-16	16	Worker internal non-sign-extended data width
Property	COEFF_WIDTH_p	ULong	-	-	Readable, Parameter	1-32	16	Coefficient width

## 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	ZeroLengthMessages=true	Signed complex samples
StreamInterface	out	32	ZeroLengthMessages=true	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.

## Performance and Resource Utilization

### `fir_complex_sse.hdl`

#### Worker Build Configuration “0”:

Table entries are a result of building the worker with the following parameter sets:

- `ocpi_endian=little`
- `ocpi_debug=false`
- `COEFF_WIDTH_p=16`
- `DATA_WIDTH_p=16`
- `NUM_TAPS_p=64`

Table 1: Worker Build Configuration “0”

OpenCPI Target	Tool	Version	Device	Registers	LUTs	Fmax (MHz)	Memory/Special Function
stratix4	Quartus	15.1.0	EP4SGX230KF40C2	11671	9,490	N/A	DSP18x18s=256
virtex6	ISE	14.7	6vlx240tff1156-1	8436	9491	156.495	DSP48E1s=128
zynq	Vivado	2017.1	xc7z020clg484-1	8902	6766	153.304	DSP48E1s=128
zynq_ise	ISE	14.7	7z020clg484-1	8436	9491	159.693	DSP48E1s=128

## 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  $\pm 1$  difference is allowed in value while comparing output rails against the filter coefficient values. Figures 3 and 4 depict the filtered results of the impulse input.

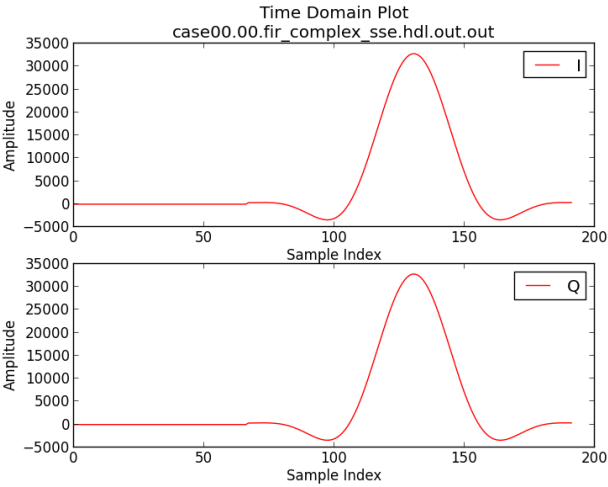


Figure 3: Time Domain Impulse Response

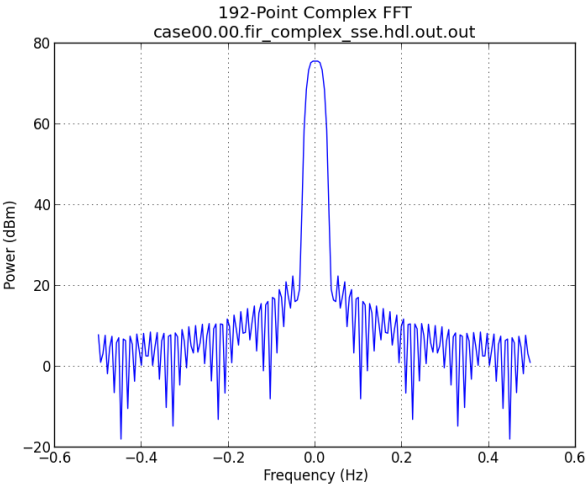


Figure 4: Frequency Domain Impulse Response