

OpenCPI  
FIR Complex SSE (TimeStamped) Component Data Sheet

OpenCPI Release: v2.4.7

## Summary - FIR Complex SSE (TimeStamped)

|                   |  |
|-------------------|--|
| Name              | fir_complex_sse_ts                     |
| Worker Type       | Application                            |
| OpenCPI Release   | v2.4.7                                 |
| Last Update       | 4/2019                                 |
| Component Library | ocpi.assets_ts.components              |
| Workers           | fir_complex_sse_ts.hdl                 |
| Tested Platforms  | isim, Matchstiq-Z1(PL), modelsim, xsim |

## Summary - FIR Complex SSE (TimeStamped)

### 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 worker also processes all operations of the Complex\_Short\_With\_Metadata protocol, passing along time and interval information. 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\_ts.hdl

The NUM\_TAPS\_p parameter defines the number of coefficient values. Care should be taken to ensure that the COEFF\_WIDTH\_p parameter is  $\leq$  the type (size) of the taps property. The taps property is type short, so COEFF\_WIDTH\_p must be between 1 and 16. Identical filter tap coefficients are applied to both real and imaginary input samples.

This implementation uses NUM\_TAPS\_p/2 multipliers for each of the real and imaginary data paths and processes input data at the clock rate - i.e. this worker can handle a new input value every clock cycle.

The FIR Complex SSE worker utilizes the OCPI *Complex\_Short\_With\_Metadata* protocol for both input and output ports. The *Complex\_Short\_With\_Metadata* protocol conveys sample data using an interface of 16-bit complex signed samples. The DATA\_WIDTH\_p parameter may be used to restrict the the number of bits processed on the input and the number of bits (sign-extended) produced on the output.

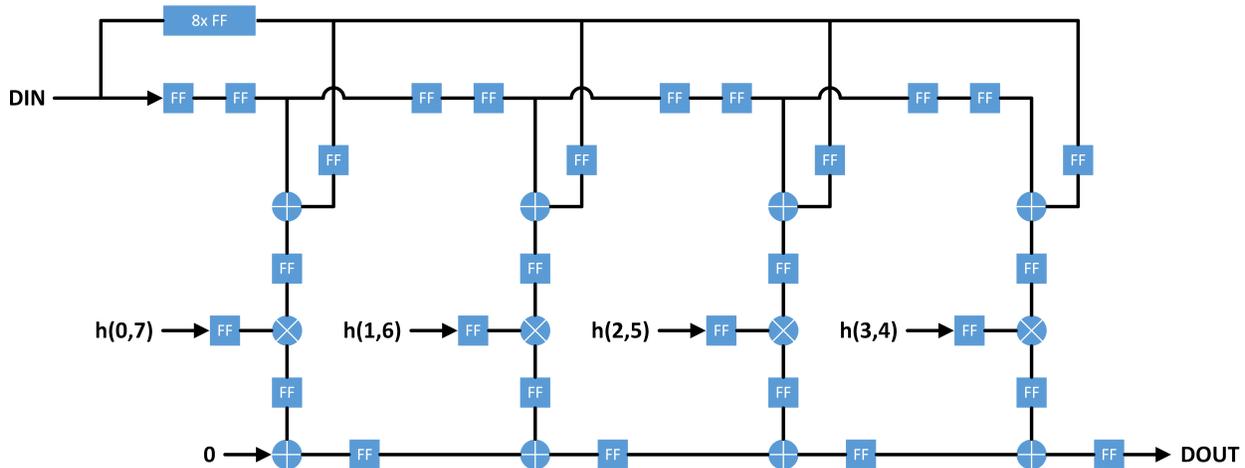


Figure 1: FIR Complex SSE Block Diagram - 8-tap example per I/Q rail

## Theory

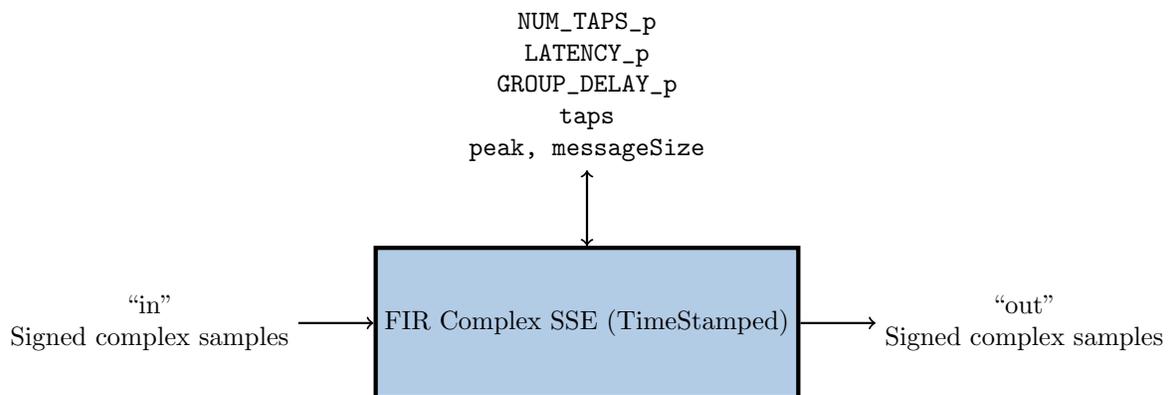
For a FIR filter with symmetric impulse response we are guaranteed to have linear phase response and thus constant group delay vs. frequency. In general, the group delay will be equal to  $(\text{NUM\_TAPS\_p}-1)/2$ . The filter topology itself will add some propagation delay to the response. For this design the total delay from an impulse input to the beginning of the impulse response will be  $\text{NUM\_TAPS\_p}/2 + 4$  samples.

The worker only outputs samples after the delay has occurred. During `flush` or `done` operations, the worker continues to produce valid data until the pipeline is empty. During the `sync` operation, the pipeline is emptied, but no valid data is produced. During `flush`, `done`, and `sync` operations, the operation is passed along after the pipeline is empty.

The worker passes along and does not modify `time` and `interval` operations.

## Block Diagrams

### Top level



## Source Dependencies

### fir\_complex\_sse\_ts.hdl

- projects/assets\_ts/components/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 | ULong | -              | -               | Parameter     | 1-?  | 16      | 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               |
| taps       | Short | -              | NUM_TAPS_p      | Writable      | $-2^{\text{COEFF\_WIDTH\_p-1}}$ to $+2^{\text{COEFF\_WIDTH\_p-1}_1}$ | -       | Symmetric filter coefficient values loaded into both real/imag filters |

## Worker Properties

### fir\_complex\_sse\_ts.hdl

| Type     | Name          | Type   | SequenceLength | ArrayDimensions | Accessibility | Valid Range | Default | Usage   |
|----------|---------------|--------|----------------|-----------------|---------------|-------------|---------|---|
| Property | DATA_WIDTH_p  | -      | -              | -               | Parameter     | 1-16        | 16      | Number of bits of input data which are processed by FIR primitive           |
| Property | COEFF_WIDTH_p | -      | -              | -               | Parameter     | 1-32        | 16      | Number of bits of taps property values which are processed by FIR primitive |
| Property | LATENCY_p     | UShort | -              | -               | Parameter     | -           | 1       | Clock cycle delay between input and output                                  |
| Property | GROUP_DELAY_p | -      | -              | -               | Parameter     | -           | 1       | Number of clocks between first valid input and first valid output           |

## Component Ports

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

## Worker Interfaces

### fir\_complex\_sse\_ts.hdl

| Type            | Name | DataWidth | Advanced | Usage                  |
|-----------------|------|-----------|----------|------------------------|
| StreamInterface | in   | 32        |          | Signed complex samples |
| StreamInterface | out  | 32        |          | 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\_ts.hdl

## Performance and Resource Utilization

fir\_complex\_sse\_ts.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/2 filter coefficients. Input data is generated by first creating a \*.dat input file containing all of the opcodes of the Complex\_Short\_With\_Metadata protocol in the following sequence:

1. Interval
2. Sync (this opcode is expected after an Interval opcode)
3. Time
4. Samples (impulse with length numtaps\*2)
5. Samples (impulse with length numtaps\*2)
6. Flush
7. Samples (impulse with length numtaps\*2)
8. Sync
9. Samples (impulse with length numtaps\*2)

The samples messages consist 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 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.

The worker will pass through the interval and time opcodes. The samples opcode followed by flush or done will output an impulse response, showing the symmetric tap values. The samples opcode followed by sync will produce the first numtaps\*2-group\_delay tap values. In addition to the samples data, the worker also passes along the zlms.

For verification, the output file is parsed into messages. All non-samples messages should match the input exactly. The samples messages are compared to the tap values and checked to ensure they are within +/- 1.