OpenCPI
FIR Real SSE Component Data Sheet

OpenCPI Release: v2.4.7

Revision History

Revision	Description of Change	Date
v1.4		10/2018
v1.5		4/2019
v1.6	Converted Worker to version 2	11/2019
v1.7	Addition of xilinx worker Table of Worker Configurations and Resource Utilization Table removed	1/2020 5/2020

Summary - FIR Real SSE

Name	fir_real_sse
Worker Type	Application
OpenCPI Release	v2.4.7
Last Update	11/2019
Component Library	ocpi.assets.dsp_comps
Workers	fir_real_sse.hdl
Tested Platforms	CentOS7, isim, xsim

Functionality

The FIR Real SSE (Systolic Symmetric Even) component inputs real 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_real_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 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 Real SSE worker utilizes the OCPI *rstream_protocol* for both input and output ports. The *rstream_protocol* defines an interface of 16-bit real signed samples. The `DATA_WIDTH_p` parameter may be used to reduce the FIR Real SSE hdl worker's internal data width to less than 16-bits.

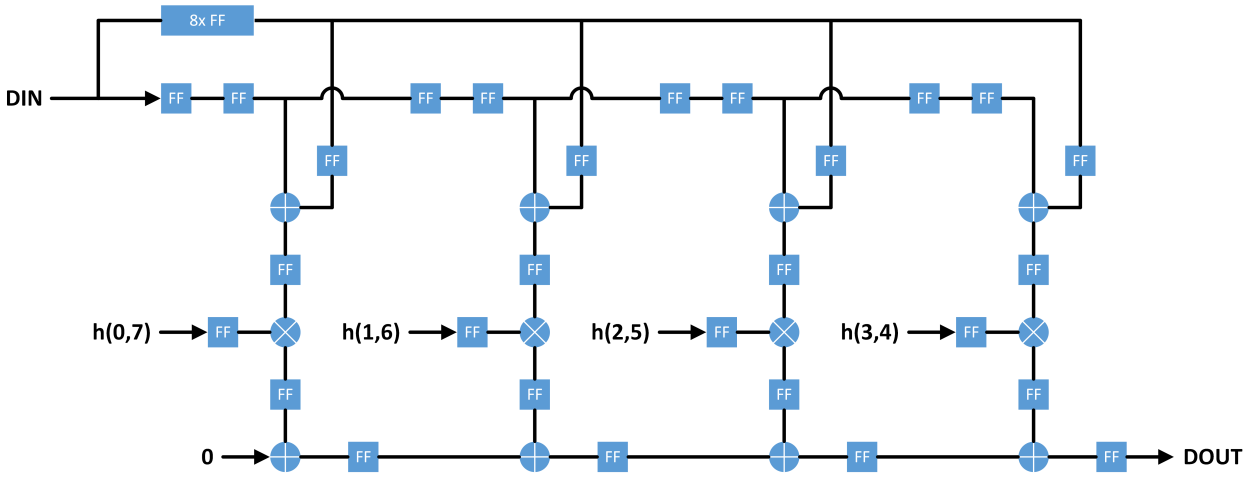


Figure 1: FIR Real SSE Block Diagram - 8-tap example

fir_real_sse_for_fskapp.rcc

The RCC implementation for the fir_real_sse_for_fskapp worker in C++ is to function as a work-alike of the fir_real_sse.hdl in the FSK app. Since the RCC worker depends upon even symmetric taps architecture and liquid API used in this worker does not provide symmetric taps, the required symmetric taps generation is done in this worker using C++ STD algorithm library. The fir_real_sse.hdl has a pipeline latency of `NUM_TAPS_p` + 4 clock cycles, and outputs `NUM_TAPS_p` + 3 zeros before the first filter output. This worker was written to function similar to fir_real_sse.hdl, and therefore has the same zeros-before-first-output behavior.

fir_real_sse_for_xilinx.hdl

The implmentation based on Xilinx IP core is generated by Xilinx FIR Compiler. The default implementation uses 4 multipliers to process input data at one sample every 25 clock cycles. If a different rate is desired a new core can be recompiled. There is a startup latency of 36 clock cycles and the output is driven directly from the core's valid signal; therefore in simulations no delay will be observed.

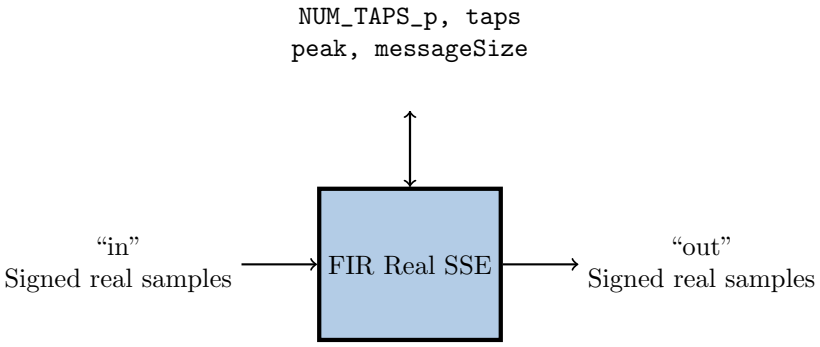
Theory

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_real_sse.hdl

- assets/components/dsp_comps/fir_real_sse.hdl/fir_real_sse.vhd
- assets/hdl/primitives/dsp_prims/dsp_prims_pkg.vhd
  assets/hdl/primitives/dsp_prims/fir/src/fir_systolic_sym_even.vhd
  assets/hdl/primitives/dsp_prims/fir/src/macc_systolic_sym.vhd
- assets/hdl/primitives/misc_prims/misc_prims_pkg.vhd
  assets/hdl/primitives/misc_prims/round_conv/src/round_conv.vhd
- assets/hdl/primitives/util_prims/util_prims_pkg.vhd
  assets/hdl/primitives/util_prims/pd/src/peakDetect.vhd

fir_real_sse_for_fskapp.rcc

- assets/components/dsp_comps/fir_real_sse_for_fskapp.rcc/fir_real_sse_for_fskapp.cc
  /opt/opencpi/prerequisites/liquid/include/liquid/liquid.h

fir_real_sse_for_xilinx.hdl

- assets/components/dsp_comps/fir_real_sse_for_xilinx.hdl/fir_real_sse_for_xilinx.vhd
  assets/components/dsp_comps/fir_real_sse_for_xilinx.hdl/reload_order_rom.vhd
  assets/components/dsp_comps/fir_real_sse_for_xilinx.hdl/gen/vivado_ip/fir_compiler_0_stub.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 even symmetric filter
peak	Short	-	-	Volatile	Standard	0	Peak value of output (valid when PEAK_MONITOR = true)
messageSize	UShort	-	-	Writable	8192	0	Number of bytes in output message (Not implemented by Version 2)
taps	Short	-	NUM_TAPS_p	Initial	$-2^{COEFF\_WIDTH\_p-1}$ to $+2^{COEFF\_WIDTH\_p-1}-1$	-	Symmetric filter coefficient values

Worker Properties

fir_real_sse.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	DATA_WIDTH_p	UChar	-	-	Parameter	1-16	16	Data Width 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.

fir_real_sse_for_fskapp.rcc

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	HDL_WORKER_DELAY	uchar	-	-	Parameter	-	4	The fir_real_sse.hdl worker has a 4 register delay.

fir_real_sse_for_xilinx.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	PEAK_MONITOR	bool	-	-	Parameter	-	true	Enable/Disable build-time inclusion of Peak Monitoring.
Property	COEFF_PADDING_p	uchar	-	-	Parameter	-	2	Numer of zero coefficients added to the beginning of filter.

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
in	false	rstream_protocol	false	-	Real signed samples
out	true	rstream_protocol	false	-	Real signed samples

Worker Interfaces

fir_real_sse.hdl

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

Control Timing and Signals

The FIR Real SSE HDL 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_real_sse.hdl

Performance and Resource Utilization

fir_real_sse.hdl

Test and Verification

Two test cases are implemented to validate the FIR Real SSE component as dictated by parameter values for `NUM_TAPS_p`: 16 and 128. In each case 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 followed by 2*(`NUM_TAPS_p`-1) zero samples. The *.bin input file is the binary version of the *.dat ASCII file repeated 2*`NUM_TAPS_p` times.

The FIR Real SSE worker inputs real signed samples, filters the input as defined by the coefficient filter taps, and outputs real 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 is simply the coefficient values.

For verification, the output file is compared against the taps file, where a ±1 difference is allowed in value while comparing the output against the filter coefficient values. Figures 2 and 3 depict the filtered results of the impulse input for the case `NUM_TAPS_p`=128.

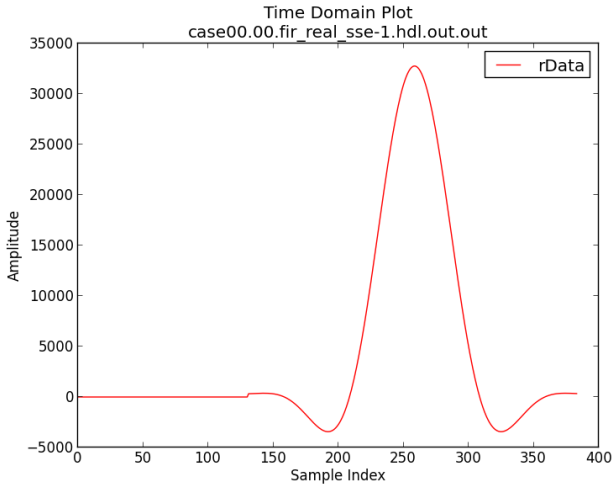


Figure 2: Time Domain Impulse Response

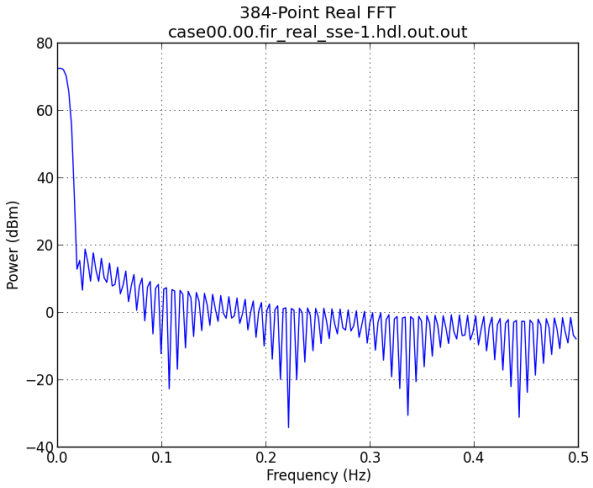


Figure 3: Frequency Domain Impulse Response