eSi-FIR

1 Contents

2 Overview

The eSi-FIR core provides an interface to filter and decimate time interleaved multi-channel data. It supports the following features:

- Programmable to operate with multi-channel data sources, e.g I/Q data.
- Programmable decimation filter rate between 1 and 4.
- Programmable number of taps between 3 and 64.
- Programmable 16-bit coefficients.
- 16-bit data
- A new filter output is available every "TAPS" clock cycles.
- Scaling and rounding of filter output to 22 bits.
- No requirement for symmetric or anti-symmetric filters
- AMBA 3 APB slave interface.
- DMA interface.

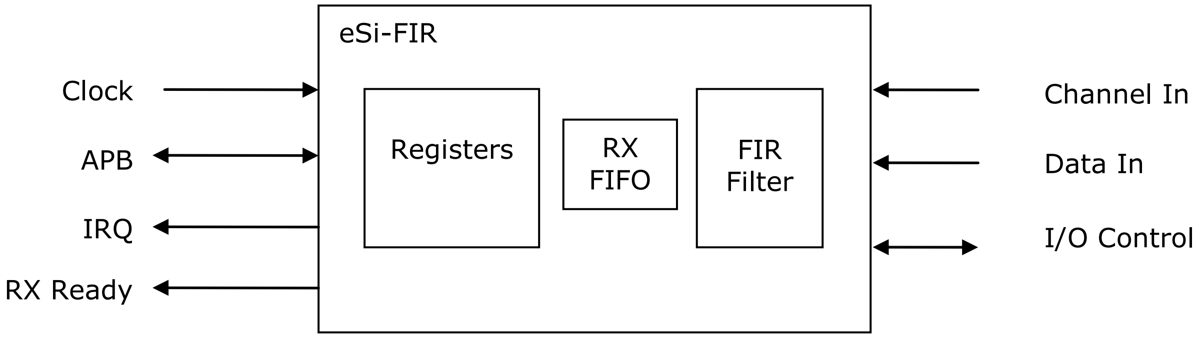

Figure 1: eSi-FIR

The control logic handles the sequencing and control for multiple channels and also decimates the filter output ready for the CPU. This reduces the number of interrupts to the processor and significantly reduces the software overhead that would otherwise be spent on filtering. The FIR filter is ideally suited for interfacing to multi-channel ADCs and takes care of conditioning the data before processing by the CPU.

3 Hardware Interface

Module Name	esi_fir
HDL	Verilog
Technology	Generic
Source Files	esi_fir.v, esi_fir_apb.v, esi_fir_filter.v, esi_fir_mac.v, esi_fir_sp_ram.v, esi_fifo.v

Parameter	Range	Default	Description
apb_address_width	3-16	16	APB address bus width
apb_data_width	16/32	16	APB data bus width (same as BITS)
tdl_data_w	8-16	16	Input data width
coef_data_w	8-16	16	Coefficient data width
coef_addr_w	2-8	6	Maximum number of coefficients $2^{coef\_addr\_w}$
ch_w	1-4	3	Maximum number of channels 2^{ch_w}
deci_w	1-2	2	Maximum decimation rate 2^{deci_w}
single_channel	0-1	0	1 if single channel input data (also set ch_w=1)
fifo_depth	2-	8	Maximum rx fifo depth

Table 1: Parameters

Port	Direction	Width	Description
pclk	Input	1	APB clock
presetn	Input	1	APB reset, active-low
paddr	Input	8	APB address
psel	Input	1	APB slave select
penable	Input	1	APB enable
pwrite	Input	1	APB write
pwdata	Input	BITS	APB write data
rx_data_in	Input	16	Input data
rx_chan_in	Input	3	Input channel
rx_valid_in	Input	1	Input valid
cactive	Output	1	Clock active
pready	Output	1	APB ready
prdata	Output	BITS	APB read data
pslverr	Output	1	APB slave error
fir_ready_out	Output	1	Ready for new input data
rx_ready	Output	1	Indicates device has data to be read
interrupt_n	Output	1	Interrupt request, active-low

Table 2: I/O Ports

For complete details of the APB signals, please refer to the AMBA 3 APB Protocol v1.0 Specification available at http://www.arm.com/products/solutions/AMBAHomePage.html

4 Software Interface

4.1 Register Map

The module is highly parameterised and the register map below is for the default configuration

Register	Address offset	Access	Description
`coefficient[0]`	0x000	R/W	Coefficient 0
`coefficient[1]`	0x004	R/W	Coefficient 1
…			…
`coefficient[63]`	0x0fc	R/W	Coefficient 63
`taps`	0x100	R/W	Taps register
`channels`	0x104	R/W	Channels register
`decimation`	0x108	R/W	Decimation register
`control`	0x10c	R/W	Control register
`status`	0x110	R/W	Status register
`accumulator_lo`	0x114	R	Accumulator result low bits
`accumulator_hi`	0x116	R	Accumulator result high bits

Table 3: Register Map

4.1.1 Coefficient Registers

The coefficient registers hold the signed FIR filter coefficients. The default configuration allows for 64 tap filters. The coefficient values may be determined by multiplying the floating point coefficients from a filter design program by 2^{15} and rounding to the nearest integer.

15 … 0
COEFFICIENT[*]

Figure 2: Format of the `coefficient[*]` register

4.1.2 Taps Register

The taps register holds the number of taps in the filter minus 1. The default configuration allows for 3 to 64 tap filters.

	5 … 0
	TAPS-1

Figure 3: Format of the `taps` register

4.1.3 Channels Register

The channels register holds the number of time-interleaved channels of input data minus 1. The default configuration allows between 1 and 8 channels.

	2 … 0
	CHANNELS-1

Figure 4: Format of the `channels` register

4.1.4 Decimation Register

The decimation register holds the decimation factor minus 1. The default configuration allows decimation by 1 to 4. The decimation works by shifting all the new input data for a channel into the tapped delay line, but only calculating the filter output and passing it the RX FIFO every `DECIMATE` input samples.

	1 … 0
	DECIMATE-1

Figure 5: Format of the `decimation` register

4.1.5 Control Register

The control register contains a selection of flags that control the operation of the module.

	2	1	0
	RF	RXIE	E

Figure 6: Format of the `control` register

Register	Values	Description
`E`	0 - Disabled 1 - Enabled	Enables the FIR interface. When disabled data will not be received
`RXIE`	0 - Disabled 1 - Enabled	Receive interrupt enable
`RF`	0 - Normal operation 1 - Reset	Reset FIFOs. When written with a 1 the RX FIFO is cleared

Table 4: Fields of the `control` register

4.1.6 Status Register

The status register contains a selection of flags that indicate the current status of the FIR interface. To clear a bit in the status register, write a 1 to it. Writing a 0 will leave it unchanged. Only the `RXO` bit may be cleared. It also contains the channel tag, which corresponds to the `rx_chan_in` channel number of the filtered sample in the accumulator. The accumulator may be repeatedly read until the `RXE` bit indicates the receiver buffer is empty.

	5 … 3	2	1	0
	CHANNEL_TAG	RXO	RXF	RXE

Figure 7: Format of the `status` register

Register	Values	Description
`RXE`	0 - Not empty 1 - Empty	Receive buffer empty
`RXF`	0 - Not full 1 – Full	Receive buffer full
`RXO`	0 - No overflow 1 - Overflow	Receive buffer overflow
`CHANNEL_TAG`	0-7	The channel number associated with the present accumulator result

Table 5: Fields of the `status` register

4.1.7 Accumulator Register

The accumulator register holds the current signed filter result. When using a 32-bit APB data bus the full accumulator result can be read in one access, whereas for a 16-bit APB data bus the register is accessed in two half-words. Reading the low order bits in the register will cause the next accumulator result to be fetched from the receiver FIFO. The result is sign extended to the full APB data bus width.

31	21	0
SIGN EXTENDED	ACCUM	

Figure 8: Format of the `accumulator` register when BITS equals 32

15	0
ACCUM_LO	

Figure 9: Format of the `accumulator_lo` register when BITS equals 16

15	5	4	3	2	1	0
SIGN EXTENDED	ACCUM_HI					

Figure 10: Format of the `accumulator_hi` register when BITS equals 16

The value of `ACCUM` is the filter result divided by 2^{15} and rounded. The accumulator has enough headroom to accommodate the full dynamic range of the result even for full-scale coefficients and data. In normal operation and for coefficients scaled such that sum(abs(coefficients)) < 1.0, then `accumulator_lo`, or the low 16-bits of `accumulator` holds the full result. When accessed by a 16-bit CPU `accumulator_hi` should be read before `accumulator_lo` if the full dynamic range result is required, otherwise it is sufficient to read only `accumulator_lo`.

4.2 Interrupts

The receive interrupt will be raised when RX FIFO is not empty and the `RXIE` flag in the control register is set to 1. This indicates that the filter has some results available for reading. Up to 8 results may be pending before the FIFO overflows.

5 FIR Filter Operation

The FIR filter incorporates a single 3 stage pipelined MAC that is efficiently time shared over the input data. The filter is capable of producing a new output sample every "TAPS" clock cycles. If the filter is set for decimation then it can accept `CHANNELSx(DECIMATE-1)` input samples on consecutive clock cycles without running the filter, followed by operating the filter `CHANNELS` times with a gap of `TAPS` between input samples. This leads to a maximum input data rate of

$$\text{maximum buffered input data rate} = \frac{CHANNELS \times DECIMATE}{CHANNELS \times TAPS + DECIMATE - 1} f_{CLK}$$

This assumes that the data source supports backpressure, for example it has a FIFO based interface. If the data source is only capable of supplying data at a constant rate then the maximum input data rate is simply

$$\text{maximum constant input data rate} = \frac{1}{TAPS} f_{CLK}$$

The control logic handles the sequencing and control for multiple channels and also decimates the filter output ready for the CPU. This reduces the number of interrupts to the processor and significantly reduces the software overhead that would otherwise be spent on filtering. The FIR filter is ideally suited for interfacing to multi-channel ADCs and takes care of conditioning the data before processing by the CPU.

By default the RX FIFO is 8 results deep to relax the interrupt service latency. The only filter control signal to the data source is `fir_ready_out`. This signal is set true when the FIR filter can accept new input data and is set false while the filter is operating. The FIR filter reads new input data when `fir_ready_out` and `rx_valid_in` are both true at a rising clock edge.

The input data memory is sized according to the maximum number of supported channels and filter taps, so in the default configuration it requires 1K bytes, whereas the coefficient memory defaults to 128 bytes.

The filter coefficients should be stored while the filter is disabled with `EN`=0. After setting `EN`=1, the first valid output is sample `TAPS/DECIMATE` and previous output samples should be read and discarded since these correspond to the time when the tapped delay line is only partially filled. Reading the filter coefficients while the filter is operating will result in corrupted accumulator data.

