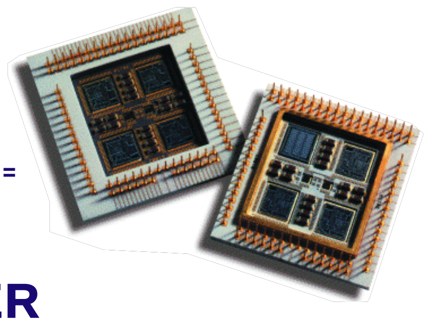
ASICentrum® A COMPANY OF THE SWATCH GROUP
Novodvorska 994, 142 21 Praha 4, Czech Republic
Tel. (+420 2) 4404 3478, Fax: (+420 2) 4149 2691, E-mail: info@asicentrum.cz



ACFIR IP CORE
FINITE IMPULSE RESPONSE FILTER

BASIC PARAMETERS

- Fully configurable fixed point FIR filter
- Two's complement arithmetic
- Pipeline architecture
- Parametrisable filter order (number of taps), data and coefficient width
- Configurable output precision
- Coefficients stored in internal ROM or RAM or dual port RAM
- Possibility to change coefficients during computation (dual port RAM)

FEATURES

Fully synchronous synthesisable RTL VHDL macro
Technological independence
Fully verified

DELIVERABLES

- Design file formats
  - RTL VHDL synthesisable code
  - Netlist
- Technical documentation
  - Data sheet
  - Implementation guide
- Verification tools
  - Testbench
- Constraints files
- Maintenance

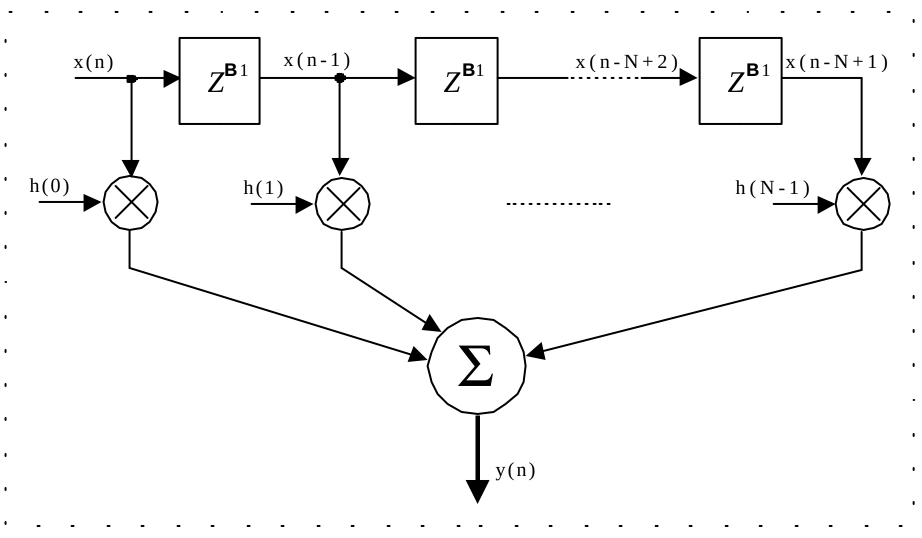
ACFIR DESCRIPTION

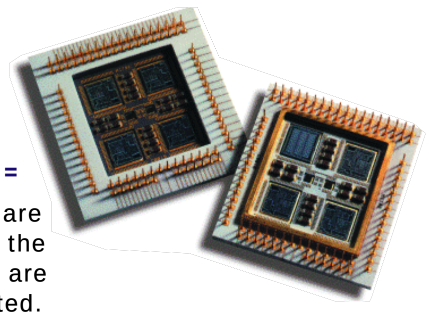
ACFIR is hardware realization of FIR filter (Finite Impulse Response filter). FIR filter is described by equation

$$y(n) = \sum_{i=0}^{N-1} x(n-i) \cdot h(i) \qquad H(z) = \sum_{i=0}^{N-1} h(i) \cdot z^{-i}$$

where y(n) is filter output, x(n) is input, h(i) is a coefficient of filter impulse response H(z) and N is the order of a filter (number of taps). This equation is explained in the following schematic.

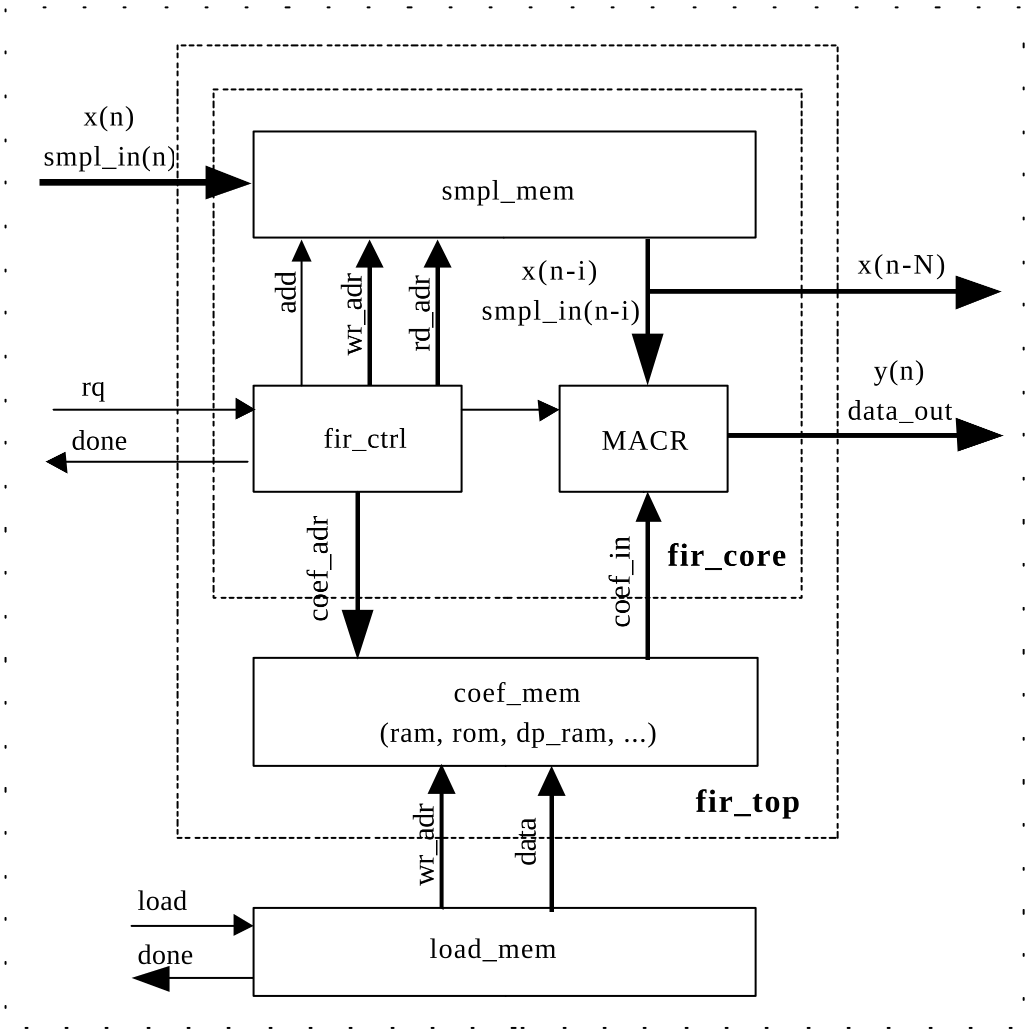
FILTER SCHEMATIC





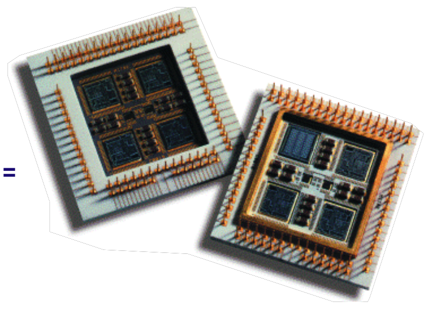
Block Z^{-1} represents time delay or memory block, where input samples are saved. When new sample x(n) comes, data from memories are shifted to the next position. The oldest sample is deleted. Samples from memories are multiplied with corresponding coefficients and these results are accumulated. Resultant number y(n) is output from the filter.
For each sample, which has to be filtered is N multiplication and N addition needed, where N is filter order (number of taps).

BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

When request signal *rq* is high, ACFIR core expects a new valid sample on the *smpl_in* port. New sample is loaded into sample memory *smpl_mem* and the output signal *done* goes low till the result on the *data_out* port is valid.
The sample memory stores N latest samples, where N is the number of taps (filter order). Sample memory is a circular buffer where the newest sample replaces the oldest one.
Data computing is done by MACR block. It consists of a pipeline multiplier, an accumulator and a pipeline round/truncate block. MACR works with saturation arithmetic and has optional number of extension bits. User can decide whether the output from the filter will be truncated or rounded per round_truncate input signal. Output flags provide condition of accumulator and round/truncate block as overflow or state of rounding, which are completely described in ACFIR pinout section bellow.
Fir_ctrl block provides computation management. This block counts adequate sample and coefficient addresses, control signals and drives output communication.
If coefficients are stored in dual port RAM, there is a possibility to change filter coefficients during computation (ideal for adaptive filtration).
In case of high filter order and non sufficient speed of ACFIR core, the filter can be, with small modifications, split into two or more filters (two or more parallel ACFIR cores) what improves almost two or more times the computation speed.



ACFIR (FIR_CORE) PINOUT

Port name	Width	Direction	Description
clk	1	Input	system clock
res	1	Input	asynchronous reset
rq	1	Input	sample processing request
round_trunc	1	Input	rounding or truncating output data
smpl_in	smpl_width_g	Input	sample data input
coef_in	coef_width_g	Input	coefficient data input
ac_near_of	1	Output	near to overflow
ac_near_uf	1	Output	near to underflow
ac_pos_half	1	Output	accumulator half full - positive
ac_neg_half	1	Output	accumulator half full - negative
ac_err_of_uf	1	Output	accumulator overflow or underflow
rl_round_up	1	Output	rounded up
rl_lim_pos	1	Output	limitation to max. positive
rl_lim_neg	1	Output	limitation to max. negative
rl_err_of	1	Output	overflow detection
done	1	Output	sample processing is done
coef_adr	addr_width_g	Output	coefficient address
data_out	out_width_g	Output	result from FIR

CORE PARAMETERISATION

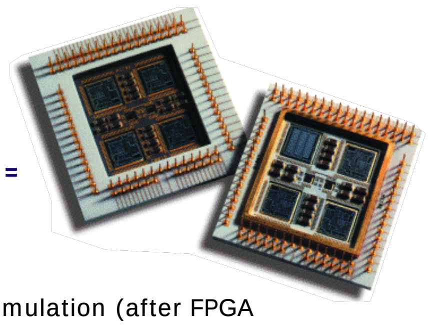
ACFIR is flexible and allows easy modification according to customer’s need.

The list of generic parameters :

Generic Name	Type	Description
addr_width_g	Natural	sample memory address bus width
smpl_width_g	Natural	sample data width
coef_width_g	Natural	FIR coefficient width
acc_ext_width_g	Natural	accumulator extension
level_g	Natural	number of stages of pipelined multiplier
out_width_g	Natural	FIR output width

FPGA XCV400BG432-4 IMPLEMENTATION OVERVIEW

CONFIGURATION	DESCRIPTION	COMPLEXITY	MAX. CLOCK FREQUENCY
Minimal	Input data width 8b Coefficients width 8b Pipeline level 3 Number of taps 64	36k gates	85 MHz
Typical	Input data width 16b Coefficients width 16b Pipeline level 4 Number of taps 128	42k gates	62 MHz
Maximal	Input data width 24b Coefficients width 24b Pipeline level 5	116k gates	63 MHz

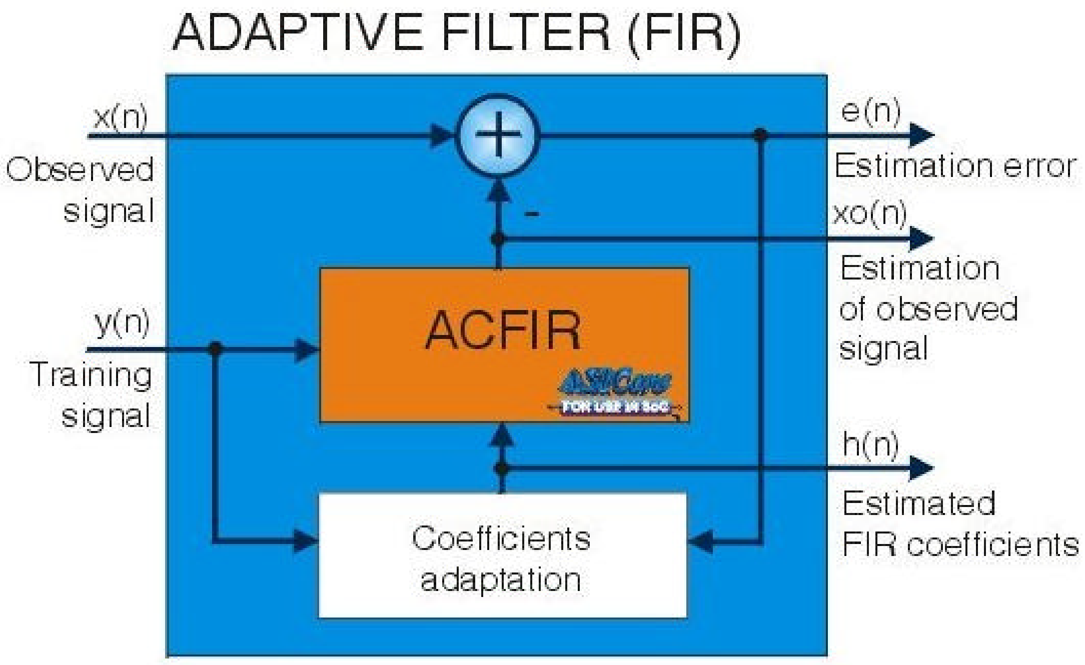


VERIFICATION METHODS

ACFIR core has been tested using functional and post-layout timing VHDL simulation (after FPGA implementation).

TYPICAL APPLICATIONS

- Linear phase low pass, high pass, band pass and band stop filters
- Anti-aliasing filters
- Interpolation filters
- Noise reduction filters
- Filters for statistical data processing
- Adaptive filtration (voice synthesis, echo cancelers....)
- Wide range of other DSP applications


ADAPTIVE FILTER (FIR)
x(n)
Observed signal
+
-
y(n)
Training signal
ACFIR
ASICore FOR USE IN SoC
Coefficients adaptation
e(n)
Estimation error
xo(n)
Estimation of observed signal
h(n)
Estimated FIR coefficients