
FFT-Based FIR (Finite Impulse Response) Filter IP Core User Manual

Unicore Systems Ltd

60-A Saksaganskogo St
Office 1
Kiev 01033
Ukraine
Phone: +38-044-289-87-44
Fax: : +38-044-289-87-44
E-mail: o.uzenkov@unicore.co.ua
URL: www.unicore.co.ua

GENERAL INFORMATION

The USFIR_FFT Application Notes contains description of the USFIR_FFT core architecture to explain its proper use.

USFIR_FFT soft core is the unit to perform the finite impulse response filter based on the Fast Fourier Transform (FFT). It performs the convolution of the unlimited signal sequence with the synthesized impulse response of the length of $M=N/2$ samples, where $N = 64, 128, 256, 512, 1024$. The data and coefficient widths are tunable in the range 8 to 18.

FEATURES

KEY FEATURES

- The filtering algorithm is the sectioned convolution with accumulating based on N -point radix-2 FFT, where $N = 64, 128, 256, 512, 1024$.
- One complex signal channel or two parallel real signal channels.
- Filter types are LPF; LPF and HPF; LPF and HPF, and differentiator; LPF and HPF, and double differentiator.
- Input data, output data, and coefficient widths are generics.
- Bandpass frequencies of the LPF and HPF filters, filter type are dynamically tunable parameters. The frequencies for both real channels are tuned independently.
- Stop band ripple for 16-bit dates is higher than 60 db. The transitional frequency band is less than 6 bins (1 bin = F_s/N , where F_s is the sampling frequency).
- Dynamic range for 16-bit dates is higher than 70 db.
- Structure optimized for Xilinx Virtex2™, Virtex4™, Spartan3™ FPGA devices, and can be implemented in Altera, Actel, Lattice devices as well.
- The maximum clock frequency for Virtex4™ devices is equal to $F_{clk} = 190$ MHz, and for Spartan3E devices is equal to $F_{clk} = 80$ MHz.
- The maximum sampling frequency F_s by $N=1024$ is less than $F_{clk}/29$.
- The latent delay of the filter by $N=1024$ is equal to 1790 cycles of F_s .

DESIGN FEATURES

Small hardware volume

The USFIR_FFT core is intended for the signal filtering with the FIR filter of large impulse response length which exceeds up to $M_i = 512$ samples.

Consider the sequential-parallel FIR filter of the length $M_i = 512$ based on the DSP48 module of the Virtex4 device with the slow-down ratio 32. This ratio for USFIR_FFT core cannot be less than 29. Then the FIR filter hardware volume occupies $512/32 = 16$ DSP48 modules and buffer RAMs. I.e. such a filter occupies in 4 times more DSP48 modules than the USFIR_FFT core does. Considering that the USFIR_FFT core has 2 parallel independent channels, the hardware volume effectiveness ratio increases to 8 times.

Comparing to the Virtex2, and Spartan3 devices, the hardware volume effectiveness of the USFIR_FFT core is higher because the DSP48 modules are absent in them.

Dynamically tunable band pass frequencies

In many applications the user needs the filters which band pass frequencies are tuned dynamically. They are adaptive filtering, software defined radio, ultrasound testing devices, etc. It is not easy problem to perform this mode in the usual FIR or IIR filters. This problem is usually solved by storing a set of coefficients of different filters or by calculating the new coefficient set each time on demand.

In the first situation the coefficient ROM has to be too large to provide the proper frequency tuning. In the second situation the calculating procedure is too complex to be performed in FPGA, and the tuning can waste too high time volume.

In the USFIR_FFT core the band pass frequencies are set simply as the codes of proper frequency bins. After new frequency setting the filter runs immediately, providing short and natural transitional process.

Highly pipelined calculations

Each FFT iteration dates are computed by the computational unit, called FFTDPATH, another words, data path for FFT calculations. FFTDPATH calculates the radix-2 FFT butterfly in the high pipelined mode. Therefore in each clock cycle one complex number is read from the data RAM and the complex result is written in this RAM. This mode supports the increasing the clock frequency up to 80 MHz and higher.

High precision computations

In the core the block floating point arithmetic is implemented. This means that the data array has the common exponent, and the array is normalized in the mode when the maximum data in the array occupies all the digits of the word. Such mode supports the high calculation precision. Due to this mode, 1024 – point FFT calculations for 16 bit data and coefficients give 70 db signal to noise ratio, which is at least at 20 db higher than calculations with the fixed point arithmetic give.

Combining the band pass filter with differentiators

In many applications the user needs to combine the band pass filter with differentiators. For example, in ultrasound testing devices the transducer has the integrator properties, which have to be compensated by differentiators. Therefore the system needs to put band pass filter and one or two differentiators sequentially. In this situation the USFIR_FFT core is the best solution because this mode is implemented in it naturally without additional hardware.

Additional frequency measurements

Often the user needs to investigate the input signal spectrum, for example, to find out the noisy frequency bands. To implement this feature the USFIR_FFT core has additional

output for signal spectrum samples or bins. This output is attached/detached on demand when instantiating the core.

FILTERING ALGORITHM

One channel real signal filter

The sectioned convolution algorithm is used for the one channel complex signal filtering. Consider $N = 1024$. This algorithm for convolution of the signal a with the impulse response h looks like the following.

- Input signal is divided into segments a_k of the length 512.
- The working array a of the length 1024 is formed as the concatenation of this segment and previous one: $a = \langle a_{k-1}, a_k \rangle$.
- FFT of the length 1024 for the working array is implemented: $A = F(a)$.
- FFT of the length 1024 for the impulse response is implemented: $H = F(h)$; note that more than a half of the array h has to be zeroed.
- The signal spectrum and the impulse response spectrum (frequency response) are multiplied: $A * H$.
- Inverse FFT of the length 1024 is derived: $y = F^{-1}(A * H)$.
- 512 resulting samples are selected which are not inferred by the circular convolution effect: $y_k = \{y_p, \dots, y_{p+511}\}$, $p = 256$.

The following considerations have to be mentioned. The impulse response h may not be transferred into the frequency response H . Instead the frequency response H can be generated due the parameters of low pass frequency F_l and high pass frequency F_h . It has to be symmetric one and has more than 512 zeroed samples.

The initial algorithm is true for the signals, which are represented by the sum of sinusoids which periods are the fractions of the FFT period. If the signal is of common form then it could not be filtered precisely by this algorithm due to the frequency aliasing effect. To minimize this effect the input signal has to be multiplied by some time window W . The resulting filtering algorithm for the real input signal is represented by the diagram on the Fig.1.

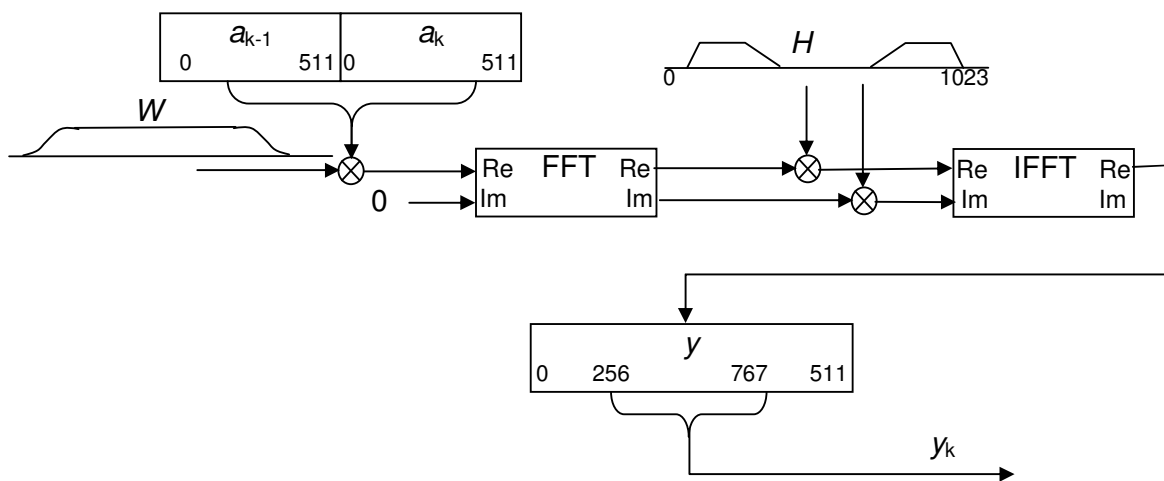


Figure 1. The filtering algorithm for a single channel

Two filters for a single real signal

When filtering a single real signal with two different filters the input signal spectrum is just the same for both filters. But the frequency response H_2 of the second filter differs from the frequency response H_1 of the first filter. To minimize the algorithm complexity the spectrum symmetry is used. If we have the real signal y_1 with the spectrum $(Y_{R1} + jY_{I1})$ on the real input of FFT, and the real signal y_2 with the spectrum $(Y_{R2} + jY_{I2})$ on the imaginary input of FFT, then after FFT we get the spectrum:

$$\begin{aligned} Y_R &= Y_{R1} - Y_{I2}; \\ Y_I &= Y_{I1} + Y_{R2}; \end{aligned} \quad (*)$$

Therefore if the spectrum of both signals is precalculated according to (*), then after IFFT we get one signal as the real part, and another signal as the imaginary part of the result. The resulting algorithm diagram is shown on the Fig.2.

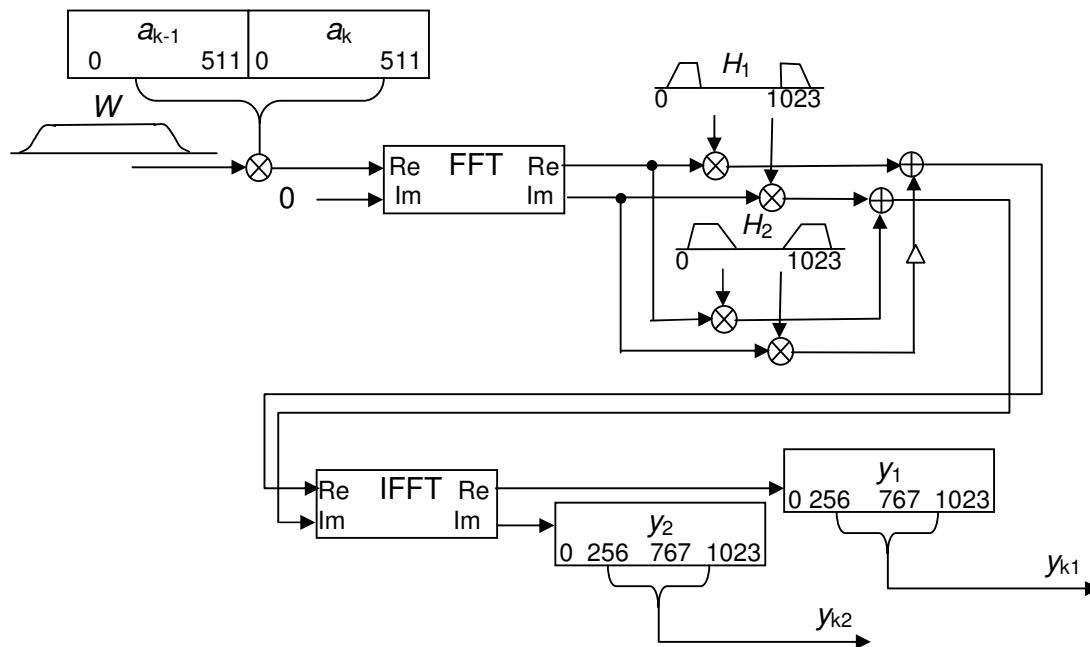


Figure 2. Algorithm of two filters for a single real signal

Two filters for two real signals

The filtering of a single input signal is performed with the abundance of operations because the imaginary part of the input data is zeroed. This abundance is minimized when the imaginary part of FFT is data of another input signal (second channel). I.e. the FFT input x is formed as:

$$x = a + jb, \text{ where}$$

$$a = \langle a_{k-1}, a_k \rangle, \quad b = \langle b_{k-1}, b_k \rangle.$$

After FFT the spectres of channels are restored from the spectrum X due to the formulas:

$$A_{Ri} = (X_{Ri} + X_{R(1024-i)})/2;$$

$$A_{Ii} = (X_{Ii} - X_{I(1024-i)})/2;$$

$$\begin{aligned}
 B_{R\ i} &= (X_{li} + X_{l(1024-i)})/2; \\
 B_{l\ i} &= - (X_{Ri} - X_{R(1024-i)})/2; \quad i=1,2,\dots,511; \\
 A_{R\ 0} &= X_{R0}; \\
 B_{R\ 0} &= X_{l0}; \\
 A_{R\ 512} &= X_{R\ 512}; \\
 B_{R\ 512} &= X_{l\ 512}; \\
 A_{l\ 0} &= A_{R\ 512}; \\
 B_{l\ 0} &= B_{R\ 512}; \\
 A_{l\ 512} &= 0; \\
 B_{l\ 512} &= 0,
 \end{aligned}$$

where *R* and *l* are indexes of the real and imaginary parts respectively.

The rest of calculations is performed in the same manner as by the filtering of a single real signal by two filters.

Differentiating

The differentiating of the real signal is equal to multiplying its spectrum at the frequency ω to the coefficient $j\omega$ ($-\pi < \omega < \pi$). By the sectioned convolution it is enough to multiply the real part of the *i*-th spectrum bin to the coefficient *i*, and the imaginary part to the coefficient $-i$, and to swap them.

Time and frequency windows

Frequency window *H* derives the selective properties of the filter. The rectangle window gives the shortest transitional frequency band. But it is bad because its IFFT has not zeros, and therefore it causes the aliasing effect.

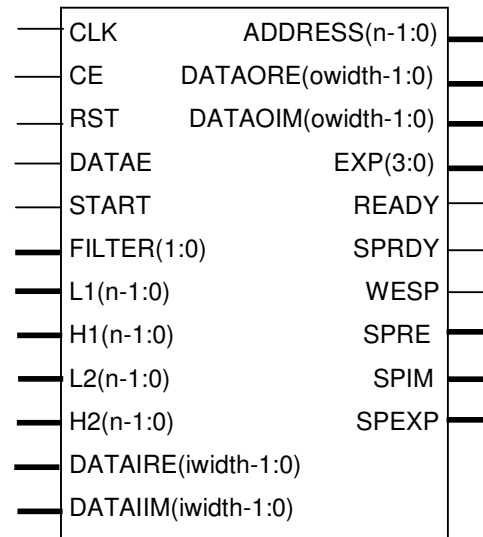
In the USFIR_FFT core the Blackman window is used which has not ripples in the band pass, and provides the suppression range more than 70 db.

The time window consists of three parts. The first and the third parts represent the halves of the Hanning window, and the second part is equal to 1.

INTERFACE

SYMBOL

Fig.3 illustrates USFIR_FFT core symbol.



USFIR_FFT_Core

SIGNAL DESCRIPTION

The descriptions of the core signals and generics are represented in the table 1.

SIGNAL	TYPE	DESCRIPTION
GENERIC S		
iwidth	natural	Input data width = 8,...,18
owidth	natural	Output and intermediate data width = 8,...,18
wwidth	natural	Coefficient width = 8,...,6
n	natural	FFT length code: 6 – 64, 7 – 128, 8 – 256, 9 – 512, 10 – 1024
reall	natural	0 – complex, 1 – real input and output signals
SIGNALS		
CLK	input	Global clock
RST	input	Global reset
START	input	Filter start
DATAE	input	Input data enable strobe
FILTER	input	00 – without filtering, 01 – LPF , LPF+HPF, 10 – LPF+HPF+ differentiator, 11 – LPF+HPF+ double differentiator
L1	input	Low band pass frequency of the first filter
H1	input	High band pass frequency of the first filter
L2	input	Low band pass frequency of the second filter
H2	input	High band pass frequency of the second filter
DATAIRE [iwidth-1:0]	input	Input data real sample (first channel)
DATAIIM [iwidth-1:0]	input	Input data imaginary sample (second channel)
READY	output	Result ready strobe
DATAORE [owidth-1:0]	output	Output data real sample (first channel)
DATAOIM [owidth-1:0]	output	Output data imaginary sample (second channel)
SPRDY	output	Spectrum start output impulse
WESP	output	Spectrum sample strobe
SPRE[owidth-1;0]	output	Spectrum real part sample
SPIM[owidth-1;0]	output	Spectrum imaginary part sample
FREQ	output	Spectrum bin number
SPEXP[3:0]	output	Spectrum data block exponent

Table 1. USFIR_FFT core signal description.

DATA REPRESENTATION

Input and output dates are represented by *iwidth* and *owidth* bit two-th complement complex integers, respectively. The spectrum data block exponent is 4-bit positive integer *e*, and the spectrum result *Y* is equal to $Y=Y_m \cdot 2^e$, where Y_m is the real or imaginary part of the spectrum data. The exponent is the same for each sample of the result array.

The code of the band frequency is equal to the bin number where the filter pass level is equal to –3 db. Codes L1,L2 have to be less than respective codes H1,H2. For instance, for $F_s = 2500$ kHz, $N=1024$, and LPF with the bandpass 400 kHz the code $H1=164$ because $400 \cdot 1024 / 2500 = 163.84$.

If $L1 = 0$ or $L2 = 0$ then the respective HPF is detached.

TIMING DIAGRAMS

On the fig. 4 the input signal timing diagrams are shown. At the start of the filter operation the signal START is inputted, which starts the core algorithm. The clock CLK frequency derives the throughput of the core. This throughput can be decreased by the periodic signal CE. And the throughput decrease rate (core activity) is equal to the rate of CE impulse width to this impulse period. In the usual mode CE=1.

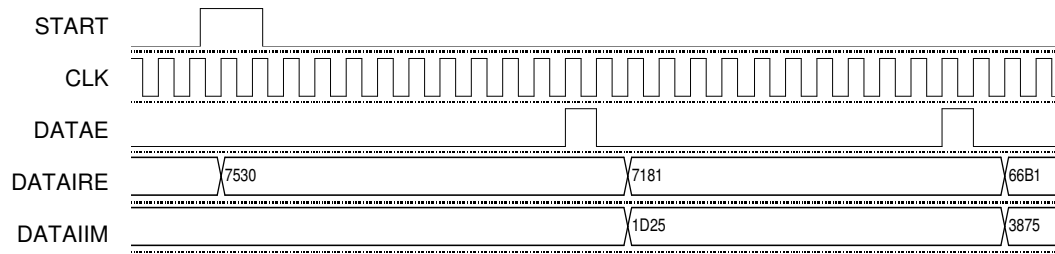


Figure 4. Input signal timing diagrams by $F_{clk}/F_s = 12$

Signal DATAE strobes the input dates and its frequency is equal to the sampling frequency F_s of the input signals DATAIRE and DATAIIM. Its width has to be equal to the clock signal period or the signal CE period. Signals DATAE, DATAIRE and DATAIIM have the setting time before the clock signal edge not less than 1 ns (is proved automatically by the synthesis).

By $N = 1024$ the sampling frequency F_s has to be in 29 times or more less than the clock frequency. By another values of N this ratio can be less than 29.

The control signals FILTER, L1, H1,L2,H2 are sampled by the core control unit and must be stable for the period of $2N$ clock impulses after the impulse SPRDY.

The timing diagrams of the output signals DATAORE and DATAOIM are represented on the fig.5. The impulse READY shows the beginning of the array output from the inner buffer memory. The strobe DATAE points to the outer device the period when the output dates are ready to be latched.

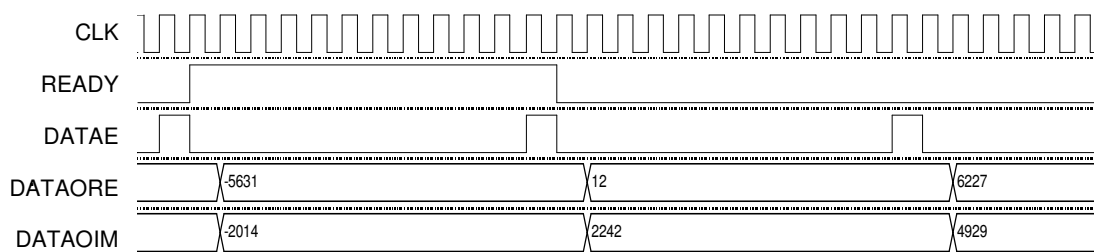


Figure 5. Output signal timing diagrams by $F_{clk}/F_s = 12$

The input signal spectrum is outputted to the outputs SPRE, SPIM as the real and imaginary parts. The spectrum timing diagrams are shown on the Fig. 6. The impulse SPRDY shows the beginning of the spectrum array output. The signal WESP=1 shows the time period of this array output. The signal FREQ is equal to the bin number. Two spectres are outputted simultaneously. In even clock cycles the spectrum bins of the first channel are outputted, and in odd clock cycles the bins of the second channel do. Because the spectrum of the real signal is symmetric one then only

the first half of the spectrum is outputted. The bins of the first channel are numbered by $FREQ$ as $0, 1, \dots, N/2-1$, and the bins of the second channel - $N/2, N/2+1, \dots, N-1$.

The signal $SPEXP$ is equal to the exponent of the spectrum array, which is common for all the bins of FFT for a single input array. To derive the correct fixed point value of bins they have to be shifted right to $SPEXP$ bits.

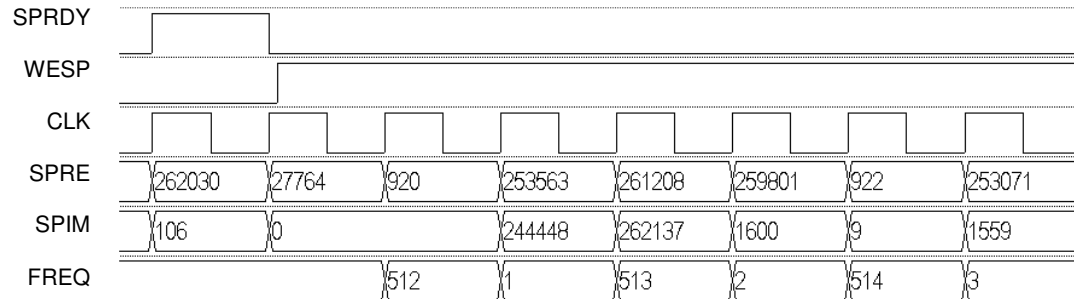


Figure 6. Spectrum signal timing diagrams

CORE STRUCTURE

USFIR_FFT core consists of the FFT processor (file ALFFT_Core_slip.vhd), IFFT processor (file ALFFT_Core_sli.vhd) and denormalizer (file DENORM.vhd) which are connected in a chain. The core structure is shown on the fig 7.

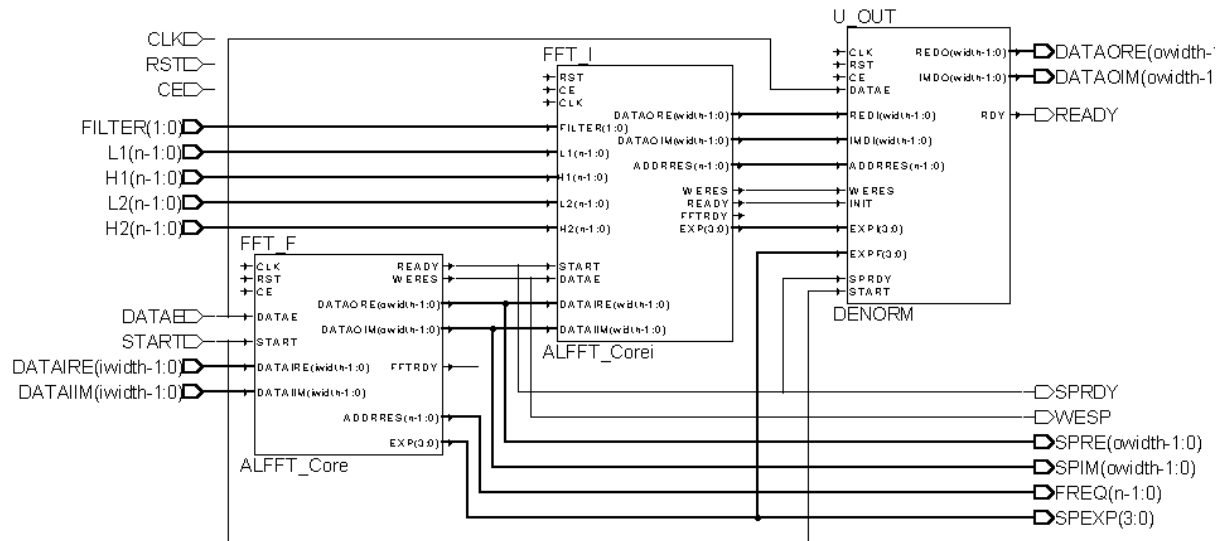


Figure 7. USFIR_FFT core structure

The input real signals of both channels enter the real and imaginary inputs of the FFT processor FFT_F . This processor performed data loading, multiplication to the time window, FFT

algorithm and spectrum restoration. At the beginning of the general cycle the inverse FFT processor FFT_I depending on the mode FILTER, performs 0,1,2,or 3 iterations of multiplication to the frequency window. On the first iteration the window is used which represents the LPF and HPF filter frequency response. On the second and third iteration the linearly increased function is used which represents the differentiator spectrum response. Then this processor implements the addition – subtraction of spectrums and inverse FFT algorithm.

The denormalizer unit U_OUT implements scaling the resulting array by right shift to the bit number which is equal to the sum of exponents derived by FFT processors. Then the scaled dates are stored to the buffer RAM, and are read from it due to the DATAE impulses.

IMPLEMENTATION

PROJECT

USFIR_FFT core is instantiated as the component in the project for FPGA using the proper EDA tools. The following files are used, which are deliverables:

- FFT_Filtr2.VHD – root file;
- ALFFT_Core_slip.vhd – FFT processor;
- FFTPATH.vhd - FFT data path;
- CONTROL.vhd - FFT processor control unit;
- ROM_COS.vhd - cosine coefficient ROM of FFT processor;
- RAM2X_2.vhd - data RAM of the FFT processor;
- ALFFT_Core_sli.vhd – IFFT processor;
- FFTPATHi.vhd - IFFT data path;
- CONTROL_i.vhd - IFFT processor control unit;
- ROM_COSi.vhd - cosine coefficient ROM of IFFT processor;
- RAM1X_2.vhd - data RAM of the IFFT processor;
- DENORM.vhd -- denormalizer;
- C.UCF – user constraint file.

The last file is formed when the project is sampled. The optimizing parameter is the minimum clock period only.

The generic constant owidth is equal to the resulting and intermediate data width. Then if the real output data width is less than owidth, then the wires have to be attached to MSBs of the outputs DATAORE and DATAOIM and the rest of bits to be open.

PERFORMANCE

The following table 2 illustrates the performance of the ALFFT core with the dual RAM block in Xilinx VIRTEX™ device when implementing 1024-point FFT for 16-bit dates and coefficients.

Target device	XC3S250E-4f	XC2VP4-7	XC4Vfx12-12
DSP48	4	4 (14%)	4 (12%)
Select Memory	10 Block RAMs	10 Block RAMs (35%)	10 Block RAMs (27%)
Area	1863 Slices (28%),	1687 Slices (56%),	1696 Slices (30%),
System clock fmax	80 MHz	150 MHz	180 MHz

Table 2. Implementation Data – Xilinx VIRTEX

VERIFICATION

To verify the USFIR_FFT project before synthesis and after it and after implementation the following files can be additionally used:

FFT_Filter2_tb.VHD – the testbench file;

RAMB4_S18_S18.vhd – behavioral model of the BlockRAM, can be substituted to the similar Unisim model.

In the testbench the USFIR_FFT core is instantiated as the component in the standard instantiation. To the core inputs the sine and cosine waves are put with the given frequency which is exchanged in time by the linear law. From the core outputs the results are sampled and analyzed.

The analysis consists in measurement of the complex vector magnitudes, their averaging and logarithm representation. The resulting signals of the testbench are:

res – result magnitude;

reslog – logarithm of the result magnitude (in decibels);

freque – sine wave frequency.

As a result, after modeling one can investigate in the VHDL simulator the frequency response of the filters by the given set of control signals. For instance, Fig.8 illustrates the frequency response of the band pass filter, and the Fig.9 illustrates the same of the band pass filter with the double differentiator.

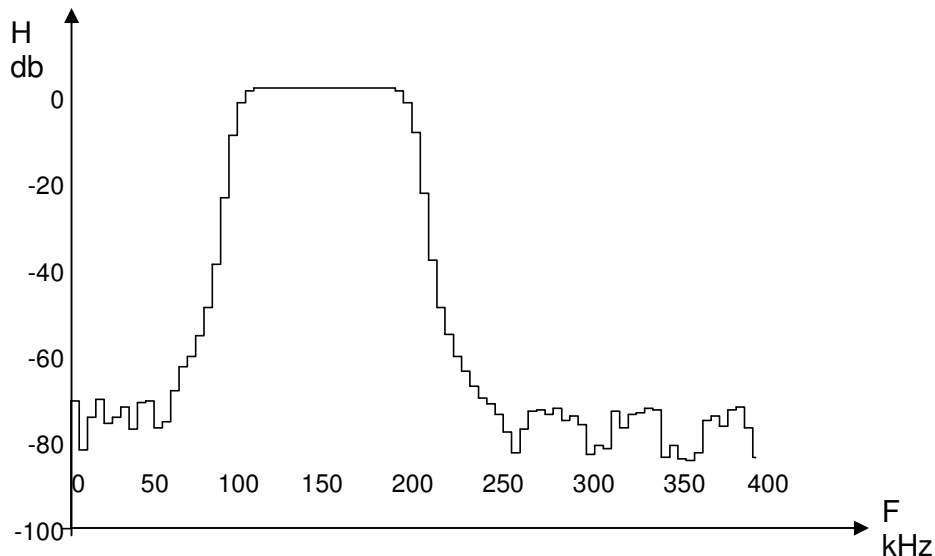


Figure 8. Frequency response of the band pass filter with the band pass 100 – 200 kHz by $F_s=2500$ kHz and $N=1024$. 1 stage width is 5 kHz.

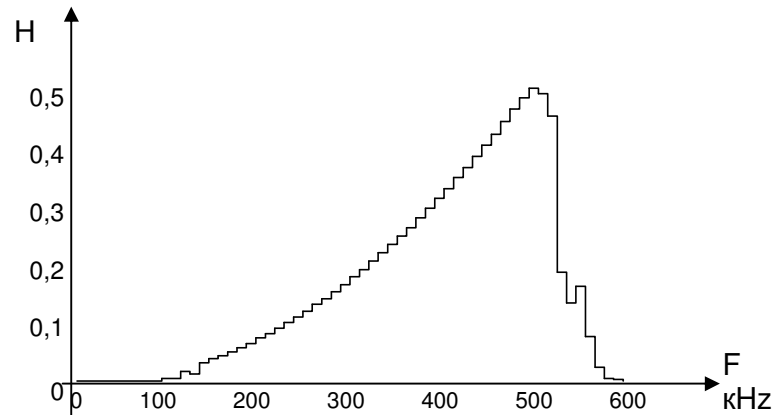


Figure 8. Frequency response of the band pass filter with the band pass 100 – 500 kHz with the double differentiator by $F_s=2500$ kHz and $N=256$. 1 stage width is 5 kHz.

The testbench has the following generic constants:

iwidth, owidth, wwidth – data widths;

n – FFT length parameter;

filtre – filter type;

tc – clock signal period;

fs – input signal sampling frequency, kHz;

nd – clock frequency to sampling frequency ratio;

df – the sine wave frequency step, kHz;

f00 – initial sine wave frequency, kHz;

magn – sine wave magnitude, 1LSB is 1;

F11,Fh1,F12,Fh2 – band pass frequencies of the filters, kHz.

To investigate the correct frequency response the band path frequencies of both channels have to be equal to each other.