DESIGN OF A LOW PASS FIR DIGITAL FILTER USING FIELD PROGRAMMABLE GATE ARRAYS (FPGAS) [+]

Ahmad Kussay Kayyali [*] **Louay Chachati*** **Mazin Rejab Khalil** [**]

Abstract:

This paper introduces a design of a low pass finite impulse response digital filter using field programmable gate arrays (FPGAs). A fully parallel distributed arithmetic FIR filter was used to attain speed/ area optimization. . A dual port (4096 x 32) RAM is designed to store the filter output samples. The different modules (elements) composing the system are assembled hierarchically to obtain a hierarchical structure design. Fixed point numbers format is used to avoid data width inflation due to successive multiplication and addition operations. Xilinx integrated software environment (ISE10.1 software) was utilized to develop the VHDL modules. The design could be applied on Xilinx Spartan3E or Vertix family. The system is tested using ISE simulator. The results obtained are compared with the results achieved by simulating the filter on DSP block set of Mat lab 7.2 version software.

تصميم مرشح رقمي ذات استجابة نبضية محددة لتمرير التردد المنخفض باستخدام بوابات المجال القابلة للبرمجة الحقلية

أحمد قصي كيالي **لؤى شاشاتي** **مازن رجب خليل غازي**

المستخلص:

تعتبر المرشحات الرقمية من أهم الأدوات الفعالة المستخدمة في تحليل الإشارات ولها تطبيقات عملية متنوعة خصوصاً في مجال معالجة الاشارة. في هذا البحث تم تصميم نظام لمرشح رقمي ذات استجابة نبضية محددة لتمرير التردد المنخفض باستخدام بوابات المجال القابلة للبرمجة الحقليةFPGA والذي يمكن استخدامة في منظومة تحويل حزمة المويجة حيث تم تصميم مرشح رقمي ذات سجلات الإزاحة المتوازية والمصممة بواسطة الحساب الموزع للحصول على تسوية مثلى بين سرعة الأداء والمساحة المستخدمة في رقاقات بوابات المجال القابلة للبرمجة الحقلية. يتطلب تصميم المنظومة ترتيب المرشحات بشكل مجزئ متعدد الأطوار نوع (1−2) لتحقيق عملية الترشيح وتنصيف عدد العينات سوية. تحتاج عينات الإخراج إلى ذاكرة لحفظها لذا تضمّن النظام تصميم ذاكرة ثنائية ألمنفذ Dual Port Memory سعة (32 x 4096) لحفظ عينات الاخراج. إن مكونات النظام تم تجميعها بطريقة متسلسلة للحصول على تصميم متسلسل لنظام المرشح كما تم استخدام الأرقام ذات النقطة الثابتة لمنع تضخم سعة بيانات الإخراج بسبب العمليات الرياضية المتعاقبة. بنيت الأجزاء المكونة لنظام التحليل باستعمال لغة وصف الكيان المادي ذات السرعة العالية

[+]**Received on 30/7/2012 , Accepted on 14/5/2013 .**
[*]**Electrical and Electronics Engineering, University of Aleppo**
[**]**Assist Lech. /Technical College/ Mosul**

(VHDL) وعلى برمجيات ISE10.1 إن النظام المصمم يمكن تطبيقه على شرائح نوع Spartan3E. تم اختبار النظام باستخدام محاكيISE ومقارنة النتائج مع النتائج المستحصلة من برمجيات Matlab7.2 وكانت متطابقة.

Introduction:

Low pass FIR digital filters have wide applications in digital communication and digital signal processing systems. Wavelet based FIR low pass filter forms an important part of wavelet packet transform technique that has wide applications in recognition and identification systems[1]. The wavelet packet transform requires the number of the filter output samples and the frequency spectrum to be halved[2]. The increasing demand for real time operation in the above mentioned fields necessitated the search for high performance hardware implementation that meets the necessary computational requirements of the FIR digital filters.

Field Programmable Gate Arrays (FPGAs) provides a suitable implementation Platform for this type of analysis due to its high flexibility and adaptability with the system design requirements[3]. Programmable logic design has entered an era in which device densities are measured in the millions of gates and system performance is measured in hundred of megahertz. Given, these system complexities, the critical success factor in the creation of design is the reasonable speed/area optimization.

Very High speed Hardware Description Language (VHDL) is widely used in designing system modules as it is standard technology/ vendor independent language[4]. Vendors usually offer a design tool suite to facilitate FPGAs system design.

The target of the research is to design a low pass FIR digital filter that can be used in wavelet packet analysis and to be mapped on FPGAs, the output of the designed filter is to be stored in a (4096x32) random access memory. The design procedure starts by explaining the theoretical bases of the digital filters, distributed arithmetic, decimating filters and memories. The procedure describes the operation of each module and how to connect the modules hierarchically to construct the system. The system is then tested to verify its functionality.

Low Pass FIR Digital Filter:

The type of low pass FIR digital filter to be designed is a Wavelet packet based filter this necessitates that the filter must halve both the frequency spectrum and output samples. In order to halve the frequency spectrum let us select a digital filter that is based on db7 wavelet. The algorithm of computing the impulse response coefficients of a wavelet based FIR digital filters depends on the basic recursion equation of a Multiresolution formulation[1]. The coefficients of the impulse response h[n] are shown in equation (1) [5] .

h[n]= [0.0004 -0.0018 0.0004 0.0126 -0.0166 -0.0380 0.0806 0.0713 -0.2240 -0.1439 0.4698 0.7291 0.3965 0.0779] ------ (1)

Where h[n] represents the impulse response of a wavelet based low pass FIR filter.

The characteristics of the impulse response is tested using the filter design and analysis (FDA) tool in MATLAB tool box, the impulse response is shown in figure 1 and the frequency response curve is shown in figure 2 from which it is noticed that the magnitude is -3db at 0.5 of the normalized frequency indicating that the filter is a half band filter.

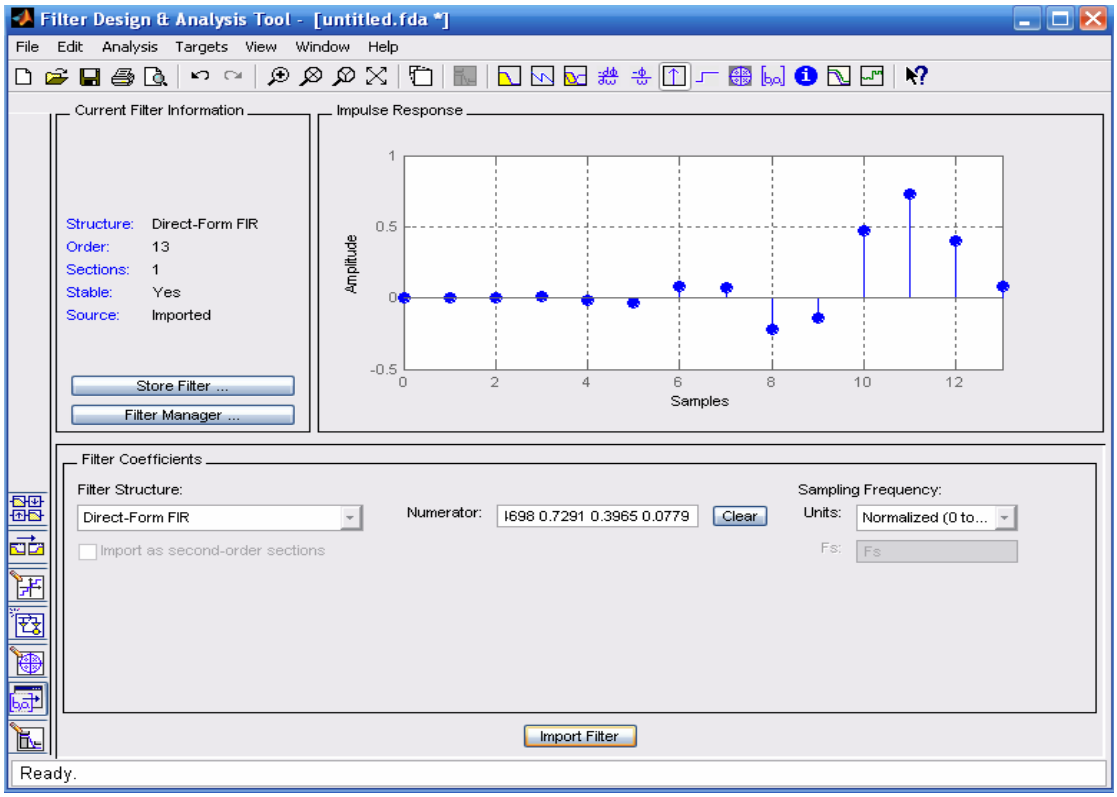


Figure 1. The impulse response of dB7 based LP FIR filter.

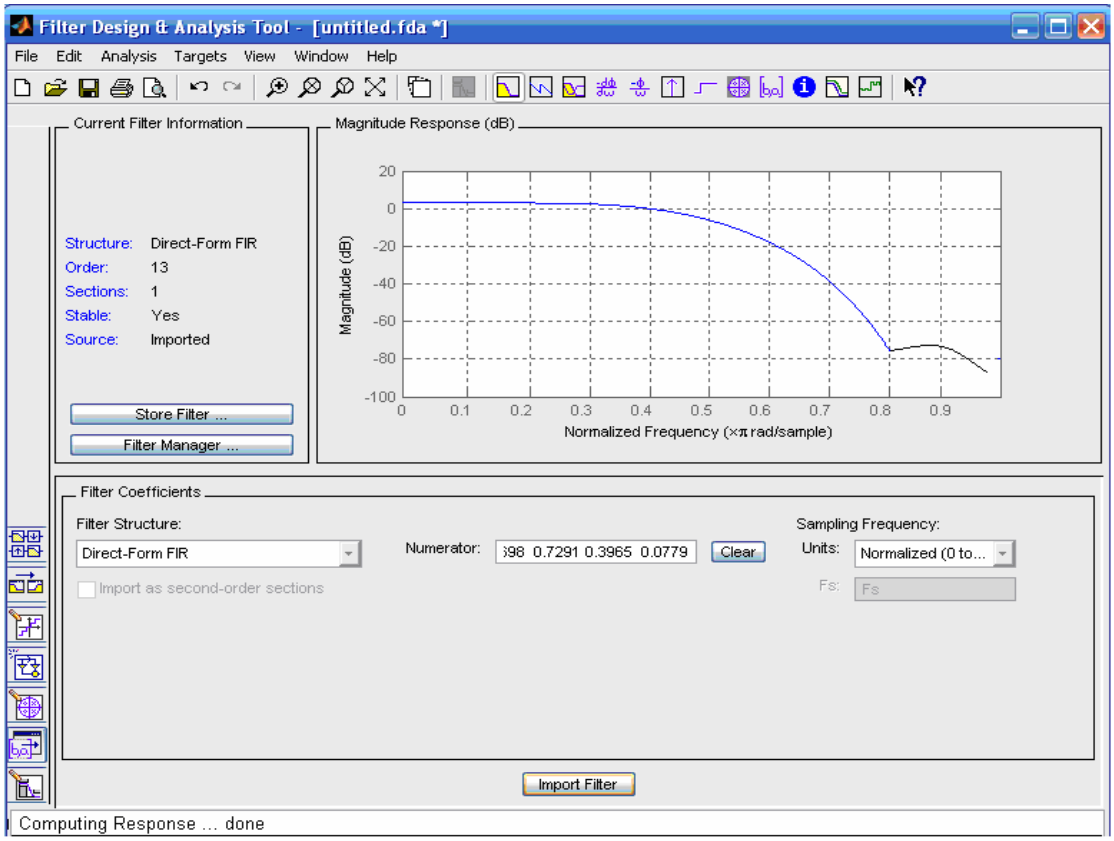


Figure 2. The frequency response of dB7 based LP FIR filter.

FIR Filter Design using Distributed Arithmetic:

A simplified view of FIR filter realization using distributed arithmetic is shown in figure 3. The basic operations required are a sequence of table look-ups, additions, subtractions and shifts of input data sequence. All these functions efficiently map to FPGAs decreasing the overall structure size of the filter [6,7].

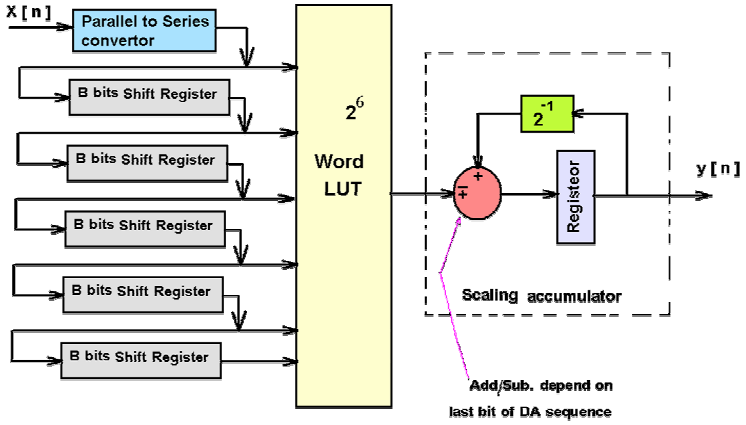


Figure 3. Six Taps FIR Filter Realization using Distributed Arithmetic.

The DA- architecture is inherently bit-serial so the input samples are at first fed into parallel to serial shift register at the input signal sample rate. As the new sample is serialized, the bit wide output is presented to a bit-serial shift register. The nodes in the cascade connection of the shift registers are used as address input to a look- up table. The way distributed arithmetic operates could be explained mathematically by rewriting the convolution equation (2) which describes the output of the filter in the form equations (3) and (4)[8]:

$$y(k) = \sum_{n=0}^{M-1} h(n)\ X(k-n) \qquad \text{k=0,1 …. N-1} \quad \text{--------- (2)}$$

$$y(k) = \sum_{n=0}^{M-1} h(n) \sum_{b=0}^{B-1} X_b[k]2^b \qquad \text{----- ---------- (3)}$$

$$y(k) = \sum_{b=0}^{B-1} 2^b \sum_{n=0}^{M-1} h(n)X_b[n] \qquad \text{---------- (4)}$$

Where
M is the number of coefficients (taps) of the digital filter.
N is the number of input samples.
X[n] is the discrete input samples.
y[n] is the discrete output samples.
If the DA filters is treating with fixed point numbers, the term 2^b will be rewritten as 2^{-b}. With the increase of filter order, the scale of the LUT will increase accordingly, which will take more time to look up the table, to solve this problem; the look up table can be divided into smaller LUT units with 4bits address. Then the values of the divided LUT units are added. Table 1 shows the coefficients and values stored in a four bits LUT [9].

Table 1. Contents of LUT

Four Bits Address	Contents	Content values
0000	0	0
0001	h[0]	0.0004
0010	h[1]	-0.0018
0011	h[0]+h[1]	-0.0014
0100	h[2]	0.0004
0101	h[0]+h[2]	0.0008
0110	h[1]+h[2]	-0.0014
0111	h[0]+h[1]+h[2]	-0.0010
1000	h[3]	0.0126
1001	h[0]+h[3]	0.0130
1010	h[1]+h[3]	0.0108
1011	h[0]+h[1]+h[3]	0.0122
1100	h[2]+h[3]	0.0130
1101	h[0]+h[2]+h[3]	0.0134
1110	h[1]+h[2]+h[3]	0.112
1111	h[0]+h[1]+h[2]+h[3]	0.1124

It is quietly obvious that DA based computations are bit serial in nature. When the input samples are presented with B bits, B clock cycles are required to complete an inner-product calculation for a nonsymmetrical impulse response. The speed of multiplication can be increased by partitioning the input words into a number of sub words (q sub words) and process these sub words in parallel. This method requires q times as many as look- up tables and so comes at of increased storage requirements. Maximum speed is achieved by factoring the input variables into single bit sub words [7,9]. The resulting structure is a fully parallel distributed arithmetic FIR filter (PDA filter), with this factoring a new output sample is computed on each clock cycle.

Consider a non symmetrical filter with 16- bit precision input samples, using serial DA filter, new output sample is available every 16 clock period. If the data samples are processed 2 bits at time, a new output is ready every 8 clocks, with processing 4,8,16 bits at a time a new output sample is available every 4, 2 and 1 clock cycles respectively. The higher the degree of filter parallelism, the fewer number of clock cycles per output sample and the greater the FPGA logic resources required to implement the design[10,11]. The designer need to trade off the area with speed.

Down sampling:

Down sampling by 2 is the process of halving the number of output samples which is achieved by using poly phase filter configuration with two phases to perform filtering and embedded down sampling by 2 to the input signal[1]. The structure is shown in figure 4.

The two phase segments are accessed by delivering the input samples X[n] to there inputs via an input commutator which starts at the segments with odd coefficients up to the segment with even coefficients. After the commutator has executed one cycle and delivered two input samples to the filter, a single output sample is taken as the summation of the outputs from the two segments. The output sample rate is half the sample rate of the input data stream.

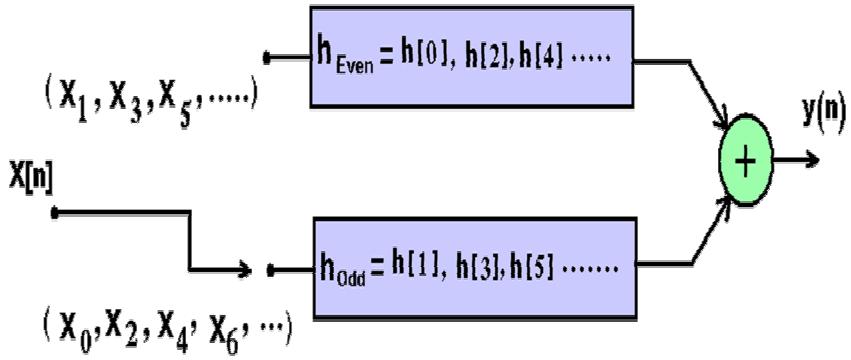


Figure4. Low Pass Decimator

If the coefficients of the two filters (h(0), h(1), h(2),… h(n)) represent low pass FIR filter, the decimator will be referred to as low pass decimator**.** Decimation is the process of low pass filtering followed by down sampling. The original signal can be reconstructed by interpolation process which is a low pass filtering preceded by up sampling operation. To illustrate the operation of the low pass decimator shown in figure 4, let us assume a low pass filter with four coefficients (h0,h1,h2,h3), its output for four input samples(x0,x1,x2,x3) before down sampling is shown in table 2. The output of the low pass decimator shown in figure 4 is illustrated in table 3, from which it is seen that the output of the decimator is the odd numbered samples of the low pass filter output before down sampling, tables 2 and 3 show that the decimator configuration of figure 4 performs filtering as well as down sampling by 2.

Table2. The output of a four coefficients low pass filter to four samples input.

n	y(n)
0	h0x0
1	h0x1+h1x0
2	h0x2+h1x1+h2x0
3	h0x3+h1x2+h2x1+h3x0
4	h1x3+h2x2+h3x1
5	h2x3+h3x2
6	h3x3

Table 3. The output of the low pass decimator with four coefficients to four samples input.

n	Even Coefficients Filter Output	Odd Coefficients Filter Output	y(n)
0	h0x1	h1x0	h0x1+h1x0
1	h0x3+h2x1	h1x2+h3x0	h0x3+h2x1+h1x2+ h3x0
3	h2x3	h3x2	h2x3+h3x2

FIR LPF Design With FPGAs :

The filter is designed in a hierarchical form using very high speed integrated circuit hardware description language (VHDL). Figure 5. shows the block diagram of the designed low pass decimator. The design is performed by partitioning the system into several blocks. Each block is designed separately, functionally tested using Xilinx ISE simulator and

connected to the previous tested block using positional mapping. The behavior of the resulting block is tested to ensure proper functionality, and so on the next block is designed and part by part the entire system is constructed and tested by behavioral simulation.
The design takes the following concepts into consideration:
a. Plausible utilization of the FPGA resources such that the number of consumed slices, flip flops, multipliers,…. etc is checked in each stage.
b. Area and speed optimization, because the optimization of time leads to an increase in the area, a compromise solution is taken into consideration.
c. Instantiation of the available intellectual properties (IP) and IP cores which were provided as part of Xilinx foundation ISE software.
d. The output samples of the filter are to be stored in a memory to be used later.

Figure 5. shows the architecture of the designed low pass decimator.

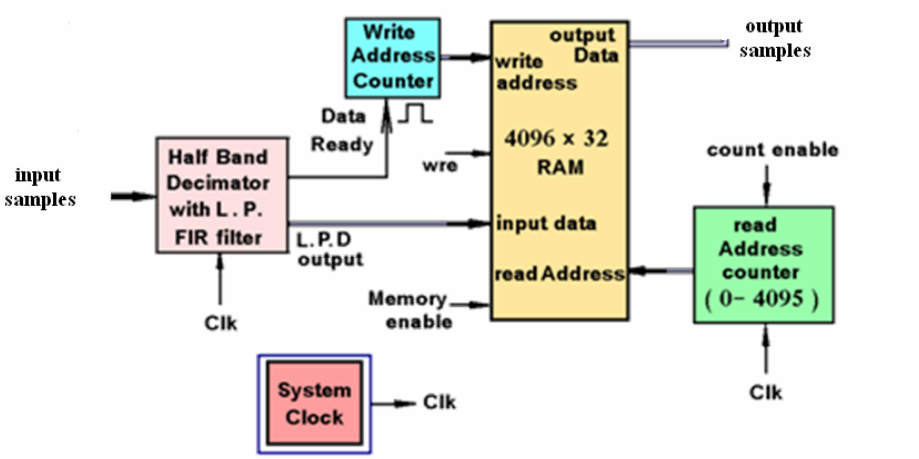


Figure 5. The architecture of the low pass decimator system.

Xilinx ISE software provides a highly parameterizable, area efficient and high performance basic DA FIR filter core. The core is generated using core generator tool available in ISE10.1 software. Figure 6. shows the register transfer logic (RTL) schematic diagram for single channel instance of this module.

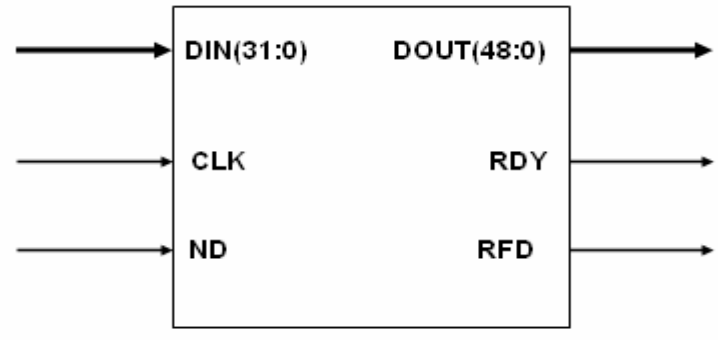


Figure 6. Schematic RTL diagram for single channel FIR filter.

Table 4. Filter parameterization table.

Parameter Name	Selected Parameters
Component Name	Low-Pass
Filter Type	Decimating Half band
Number of Taps	14
Impulse Response	Non Symmetric
Coefficient width	13 bits
Coefficient Type	Signed
Coefficients	Shown in table 2
Input Data width	32 bits
Input Data Type	Signed
Implementation Option	Parallel DA
Clock cycle/ output Sample	1
Latency	11 clocks

Table 5. Filter Coefficients.

Real Numbers	Hexadecimal Numbers
0.0004	0001
-0.0018	1ff8
0.0004	0001
0.0126	0033
-0.0166	1fbc
-0.0380	1f64
0.0806	014a
0.0713	0124
-.2240	1c6a
-0.1439	1db2
0.4698	0784
0.7291	0baa
0.3965	0658
0.0779	013e

In order to obtain the desired filter performance the user must define the input parameters as shown in table 4. The coefficients of the filter shown in table 5 are based on Daubechies wavelet coefficients(dB7) according to which the filter act as a half band low pass FIR filter[1]. As the FPGAs cannot treat with real numbers, fixed point numbers were chosen, the input samples width is 32 bits (one sign bit, eight bits for integer number and 23 bits for fractional part).The coefficients width is 13 bits(one sign bit and 12 bits for fractional part) the output samples width will be 47 bits. In order to save resources the output is truncated to 32 bits with the same format of the input samples, this necessitates building a module for truncation. In the implementation option, the selection of parallel DA specifies a fully parallel filter (PDA filter) to be produced. The clock cycles/ output sample (L) option allows the degree of filter parallelism. Selecting L=1 for half band decimator results in filter where each internal DA sub filter generates a new output sample every clock cycle irrespective of the filter input sample precision.

The filter latency is the number of clock cycles between the input data samples and the corresponding filter first output sample, this is reported in the design wizard.
Table 6 list the FIR decimator ports and their functional description.

Table 6. FIR core signals functional description

Name	Direction	Functional Description
DIN L (31:0)	Input	Decimator input data sample with 32 bit wide
Clk	Input	System clock (active rising edge)
NDL	input	New data (active high) when this signal is asserted the data sample on DIN port is loaded into the parallel to the series converter register
Dout L (46:0)	Output	Decimator output sample with 47 bit wide
RDY L	utput	Decimator output sample ready (active high)it indicates that a new filter sample is available on D out port
RFD L	Output	Ready for data (active high), it indicates that the final bit of the current data may be supplied to the filter

The decimator outputs the signal RDYL each time a new output is available at Dout, since the output at Dout is alternate output of two internal filter (as shown in figure 4), RDYL takes the form of clocks (consecutive high and low).

Random Access Memory Design :

The output samples of the half band decimator need to be stored in storage element such as random access memory. As the speed of the processing is an essential factor, a dual port RAM is designed such that, memory read operation can be arranged to be a number of clocks (it can be one clock) behind the write operation. Figure 7 shows dual port memory schematic diagram. Table 7. presents the dual-port RAM pin description.
If the input samples are selected to be 8192 samples a dual port RAMs with 4096 storage locations is to be designed, with data width of 32 bits to store the output samples of low pass FIR decimator.

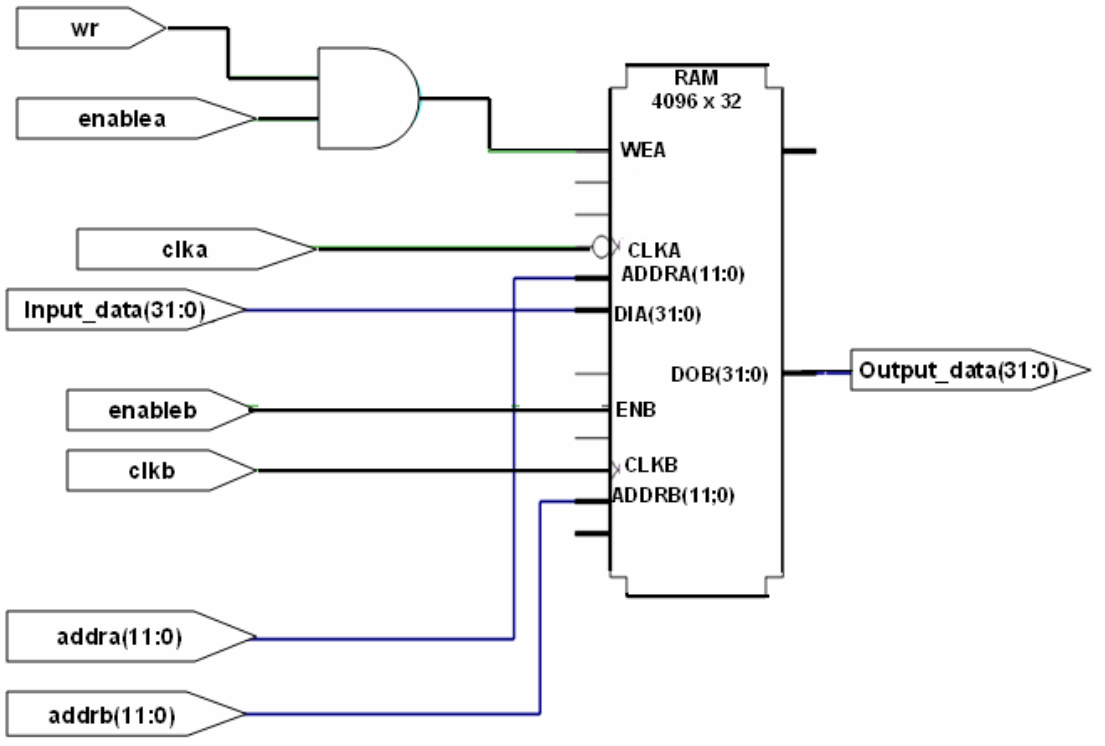


Figure 7. The RTL schematic diagram of dual port RAM diagram.

Table 7. Dual- port RAM pin description.

I/O pin	Type	Description
WEA	Active high Input	Write enable (port A)
ENA/ ENB	Active high Input	RAM Enable (port A / port B)
ADDR A	12- bit Input	Write address
CLK A	Input	Negative edge write clock
CLK B	Input	Positive edge read clock (system clock)
ADDR B	12 bits Input	Read address
DIA	32 bits Input	Data Input
DOB	32 bits output	Data output

Each of the write address ADDRA and the read address ADDRB is driven by its own 12 bit counter (0 – 4095). Figure 8 shows the counter schematic diagram, table 8 shows the pin description.

RDYL signal is used to drive the RAM write clock as well as the write address counter because RDYL signal is coincident with the decimator output to ensure that the data is correctly stored in the proper memory address.

The read address is driven by the system clock. Reading the data from the memory is controlled by enabling (enableb and ce via cer signal).

Table8. Pin description of the counter unit.

I/O pin	Type	Description
CE	Active high input	Counter enable
CLK C	Input	Negative edge clock
Count_out	12 bits output	Counter output (0- 4095)

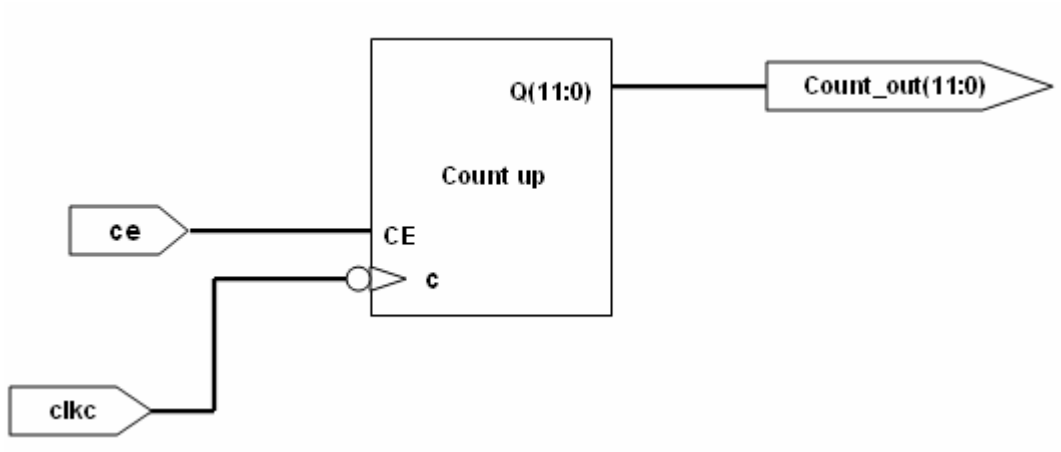


Figure 8. The RTL schematic diagram of up counter.

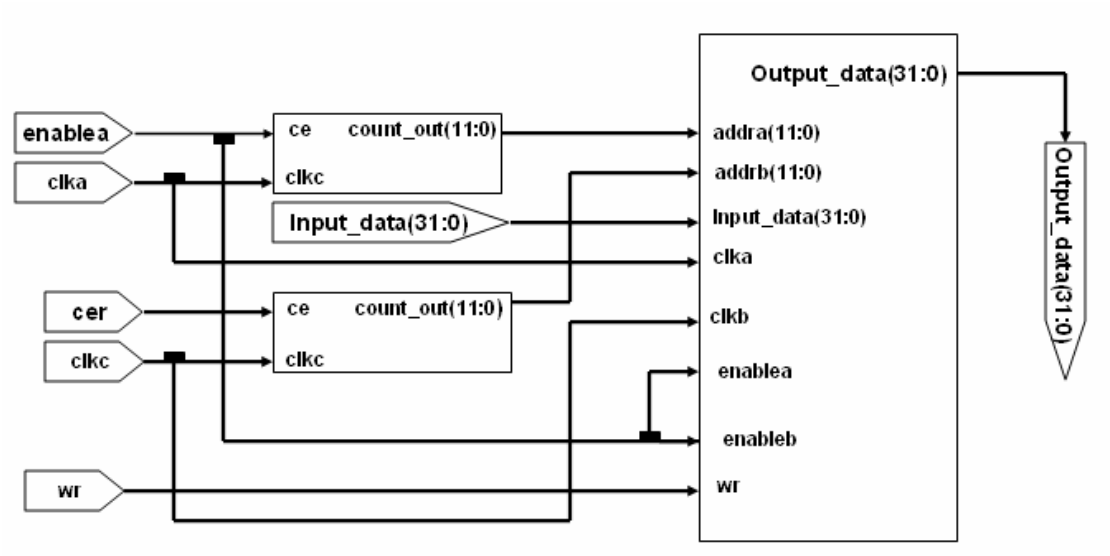


Figure 9. The RTL schematic diagram of the write and read counters connected to the memory addra and addrb respectively.

Figure 9 shows the connection of the write and read counters to (addra) and (addrb) ports of the memory. Figure 10. shows the RTL schematic diagram of the designed system. the module z11 represents the low pass decimator ,z21 module represents the memory with its counters and z31 module represents the truncation unit.

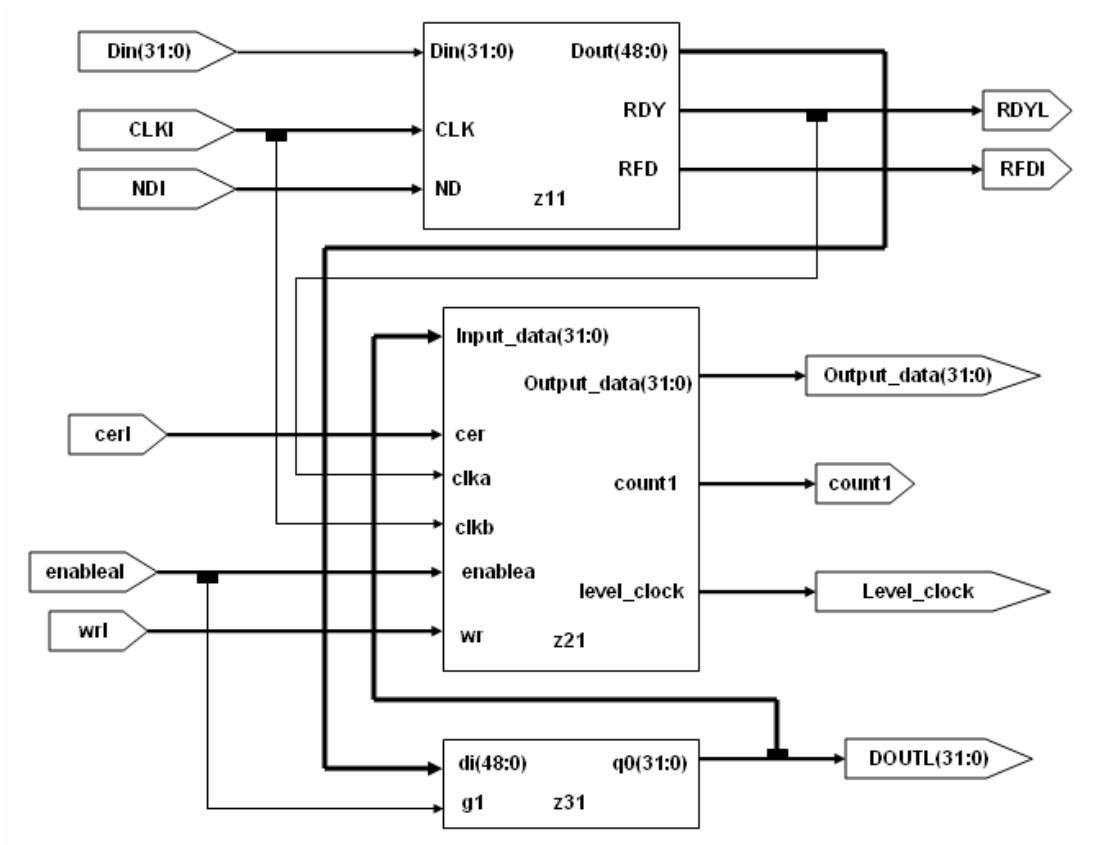


Figure 10. The RTL schematic diagram of connecting the FIR low pass decimator to the memory with its counters and truncation unit.

Practical Results:

For the purpose of simulation and results exhibition 16 samples were chosen from audio signal, and introduced to the system as input samples, the obtained results are as following:

a. Figures (11-14) displays the output of FIR low pass decimator (doutl) and output of the read operation from the corresponding RAM (output- data) from which the following notes are concluded:

- The number of output samples are (8) due to down sampling by (2).
- The number of clock cycles/ output sample is (1), indicating that the DA FIR low pass filter is fully parallel.
- The latency is 11 clocks.
- The read operation started after enabling the signal (Cer) (counter, read enable).

b. The obtained results were verified using mat lab programs. Figure 15. shows the output of the filter due to 16 samples input of an audio signal.

c. Figure 16 displays the output of the FIR low pass decimator (doutl) obtained from figure 13. transformed from hexadecimal to real number using mat lab program.

d. Comparing the results of figure 15and figure 16 displays the same results.

e. Figure 17 shows the percentage utilization of the available resources.

f. The variables, in small letters, shown on the left side of the graphs of figures 11, 12, 13 and figure 14 represent the input or output signals at the ports defined, in capital letters, in the tables 6, 7 and table 8.

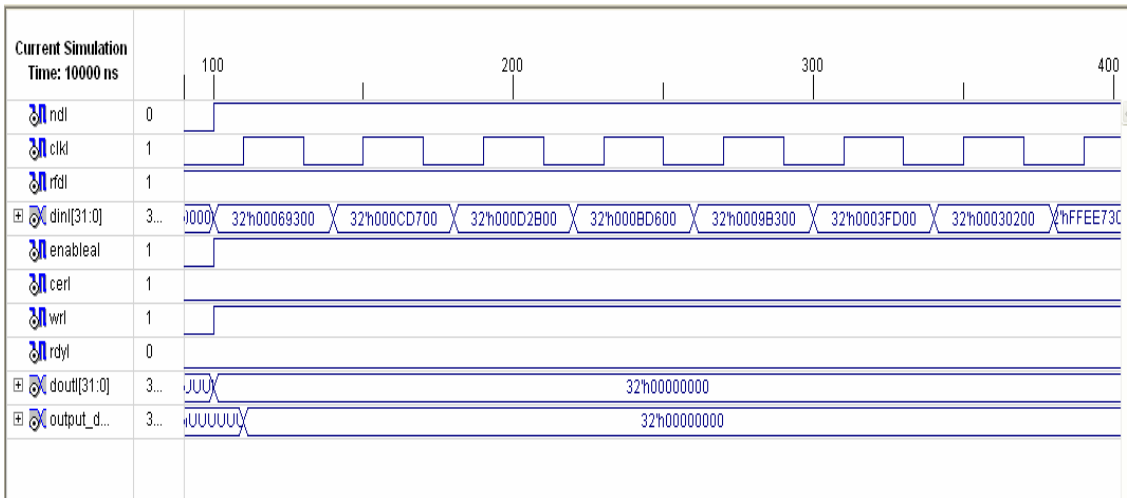


Figure 11. The input samples

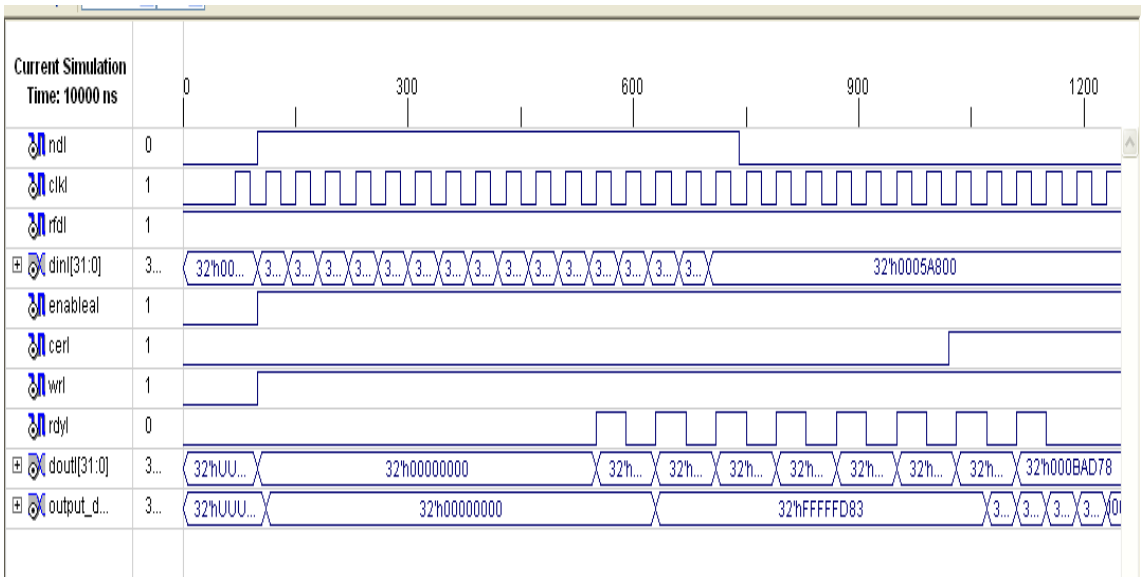


Figure 12. Control signals

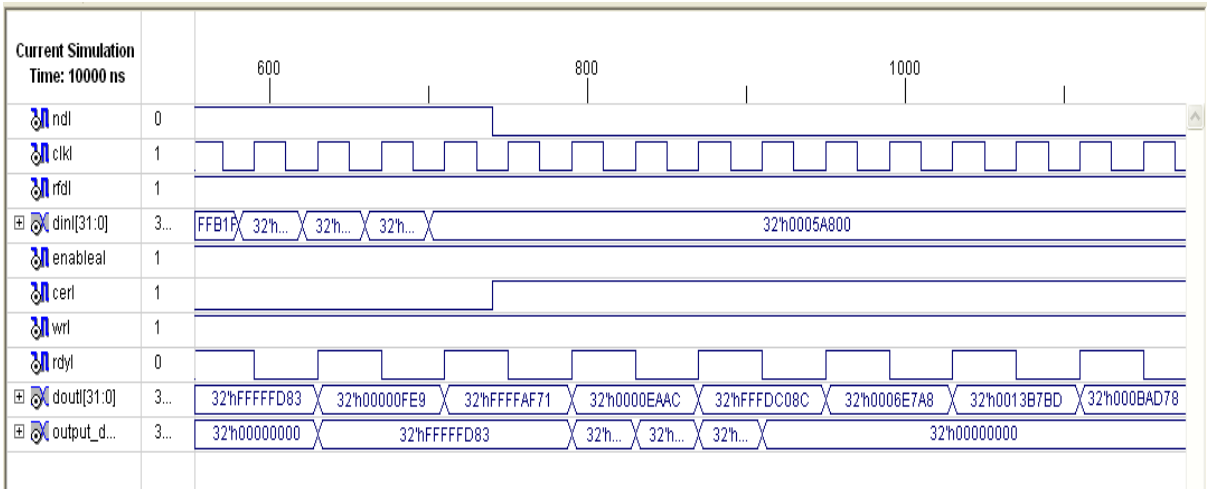


Figure 13. Filter output signals

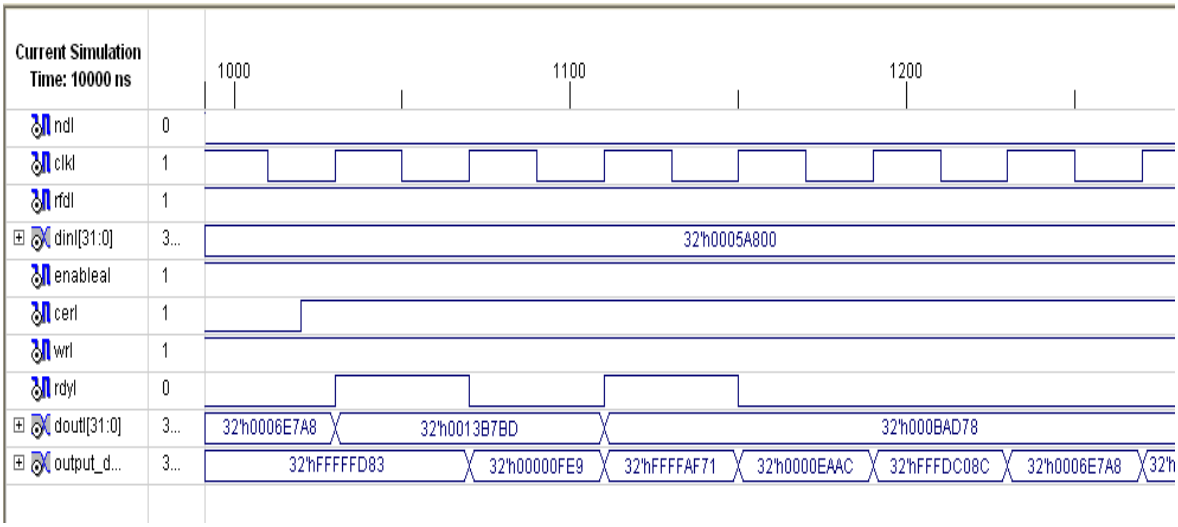
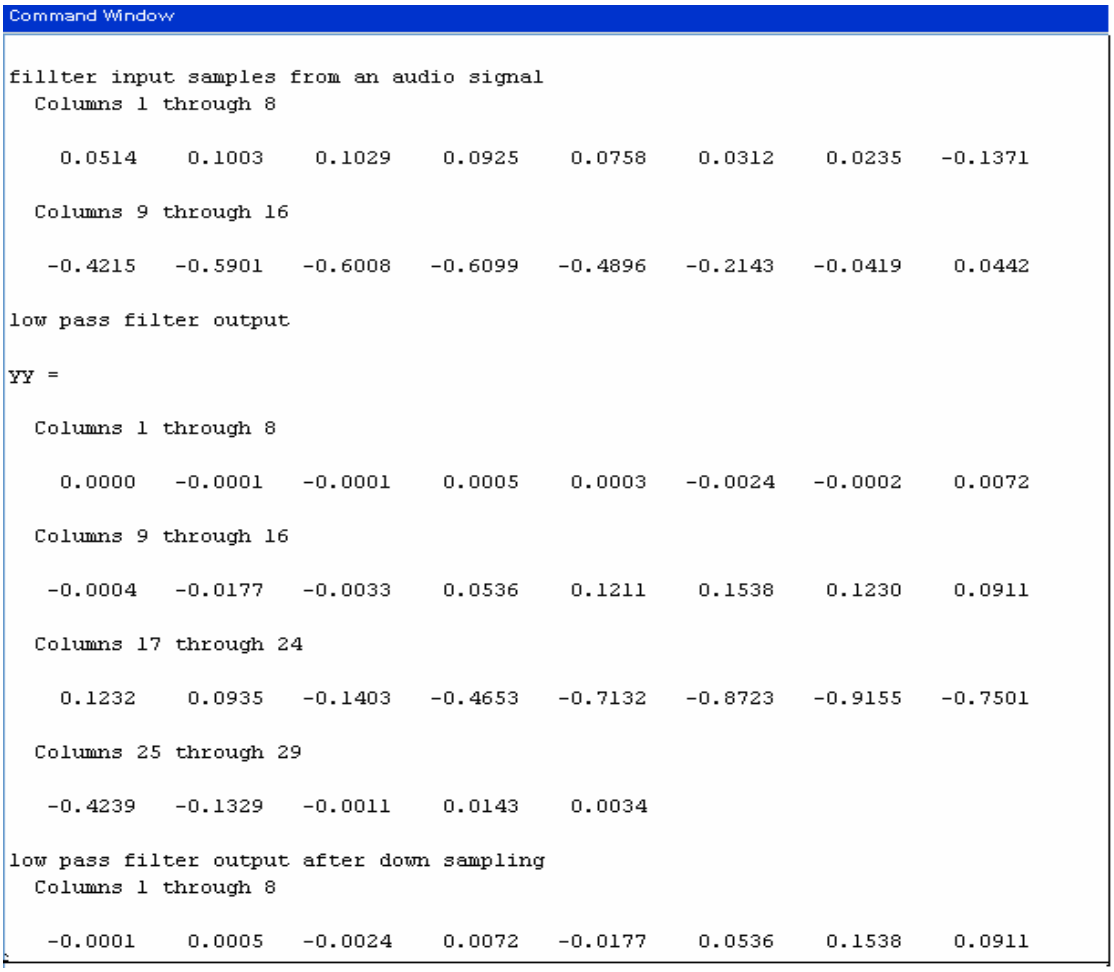


Figure 14. The filter output read from the memory.



```
Command Window

fillter input samples from an audio signal
  Columns 1 through 8

    0.0514    0.1003    0.1029    0.0925    0.0758    0.0312    0.0235   -0.1371

  Columns 9 through 16

   -0.4215   -0.5901   -0.6008   -0.6099   -0.4896   -0.2143   -0.0419    0.0442

low pass filter output

YY =

  Columns 1 through 8

    0.0000   -0.0001   -0.0001    0.0005    0.0003   -0.0024   -0.0002    0.0072

  Columns 9 through 16

   -0.0004   -0.0177   -0.0033    0.0536    0.1211    0.1538    0.1230    0.0911

  Columns 17 through 24

    0.1232    0.0935   -0.1403   -0.4653   -0.7132   -0.8723   -0.9155   -0.7501

  Columns 25 through 29

   -0.4239   -0.1329   -0.0011    0.0143    0.0034

low pass filter output after down sampling
  Columns 1 through 8

   -0.0001    0.0005   -0.0024    0.0072   -0.0177    0.0536    0.1538    0.0911
```

Figure 15. The filter output using mat lab

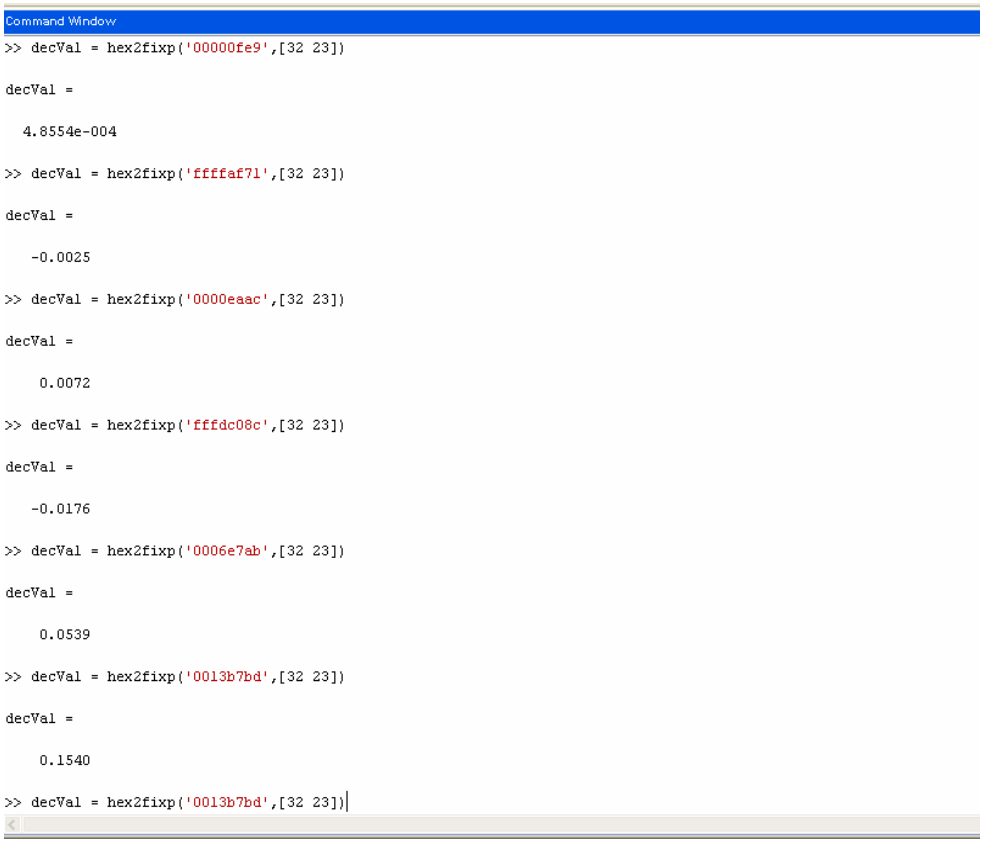


Figure 16. The filter output read from figure 13. transformed into real numbers using mat lab

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	1567	4656	33%
Number of Slice Flip Flops	2854	9312	30%
Number of 4 input LUTs	2371	9312	25%
Number of bonded IOBs	95	232	40%
Number of BRAMs	10	20	50%
Number of MULT18X18SIOs	2	20	10%
Number of GCLKs	5	24	20%

Figure 17. Device utilization summary.

Conclusions:

A hierarchical low pass FIR digital filter system is designed to be implemented on a field programmable gate array (FPGA) slice, type Spartan (3e) using a highly parameterizable, fully parallel DA FIR low pass half band filters with coefficients based on dB7 wavelet.

On the basis of the experimental results the following conclusions are inferred:

a. An efficient FPGA low pass FIR digital filter system is obtained using hierarchical designed structure. The system is area saving system because of using a fully parallel FIR filters to achieve area speed optimization.

b. The designed filter is accommodated to act as wavelet based low pass FIR by replacing the coefficients of its impulse response by the coefficients of dB7 wavelet and decimate (down sample) the out put samples by 2.
c. The inflation in the output data width due to the successive additions and multiplications operations can be eliminated by using fixed point numbers.
d. Efficient down sampling can be obtained by using poly phase filter configuration.
e. Using two dual- port RAMs in the structure of the system is necessary to save the output samples, as well as it increases data acquisition speed because it allows writing and reading data at the same time.
f. As a future work, embedded design techniques can be applied to construct the system.
g. As a future work the VHDL Code can be generated using filter design and analysis tool of MATLAB toolbox and the designed system performance can verified by using ModelSim simulator.

References:

1-BURRUS S., RAMESH A.; GUO H., *Introduction to Wavelet and Wavelet Transforms*. Prentices Hall, 1st ed, New Jersey,chap:5-6,pp50-88, 1998.
2-ALI N.; HADDAD A., *Multiresolution Signal Decomposition*. Academic Press, 2nd Ed, New York, 2002.
3-KILTS S., *Advanced FPGA Design*. John Wiley and Sons, 1st ed, New York, 2007.
4-VOLNEI A., *Circuit design with VHDL*. MIT Press, 1st ed, London, 2004.
5-CHARLES K., *An Introduction to Wavelets*. Academic Press, 1st ed, New York, 2002
6-LONNIE C., *Fundamentals of Digital signal Processing*. Harper and Row, 1st ed, New York, 1986.
7- Patrick Longa and Ali Miri, "Area Efficient FIR Filter Design on FPGAs using Distributed Arithmetic". *IEEE international Symposium on Signal Processing and Information Technology*, pp. 248-252, 2006.
8-AL-HAJ A., "Fast Discrete wavelet Transformation using FPGAs and Distributed Arithmetic". *International Journal of Applied Science and Engineering,* 1(2), 160-171, 2003.
9- Sivakumar V. S., Venkateshwarlu.K and Gopikrishnarao P.V, "Efficient Implementation of FPGA Based Distributed Arithmetic Architecture For FIR Filter". *National Conference on Emerging Trends in Information Digital& Embedded Systems*, IJCAE, Vol.3, pp 61-23, July 2012.
10- Wang Sen, Tang Bin, and Zhu Jun,"Distributed Arithmetic for FIR filter Design on FPGA". *IEEE International Conference on Communications, Circuits and Systems*, pp. 620-623, 2007.
11- Mamatha B, Ramachandram V, "Design and Implementation of 120 Order FIR Filter Design on FPGA". *International Journal of Engineering Science& Emerging Technologies,* Vol.3, Issue 1, pp. 90-97, August 2012.