ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

ANALYSIS OF SYNTHESIS ISSUES ABOUT DESIGNING DSP DEVICES

Akash Verma[1], B.S. Rai[2]

M.Tech. (Pursuing), Dept. of Electronics Engineering, MMMEC, Gorakhpur, India[1]

Head of Dept., Dept. of Electronics Engineering, MMMEC, Gorakhpur, India[2]

ABSTRACT: This paper discusses the issues related to the synthesizing the designs of DSP devices to FPGA. The high level codes used for synthesis input, are in VHDL. The central issues behind the designs are synthesizable or not are, used HDL libraries and data types. All the issues and solutions are illustrated using 32-Point Fast Fourier Transform. In the beginning, the IEEE fixed point package (fixed_pkg) is used for designing FFT-32 then whole logic is designed using single IEEE package (STD_LOGIC_1164) which is absolutely synthesizable to FPGA. For implementing DSP algorithms using 'STD_LOGIC_1164', 'real type' data structure is represented by array of bits, that is 'bit_vector'. Algorithms for real type addition, subtraction and multiplication are developed using array of bits which will fulfill the function of complex and real arithmetic. DSP algorithms implemented through this design method are complete synthesizable and can be implemented with very high degree of precision.

Keywords: DSP, FFT, FPGA, Fixed_Pkg, Radix-2 algorithm, STD_LOGIC_1164, Synthesis, VHDL, Virtex-5.

I.INTRODUCTION

This paper proposes the issues and solutions of synthesis problems of DSP designs to FPGA [6]. DSP algorithms which are designed on VHDL do not guarantee that they are synthesizable [8] to FPGA. It may be possible that these HDL designs do compile and simulate properly on HDL simulators and compilers but still it is not sure that they are completely synthesizable to FPGA [5]. There are several issues which restricts these codes from synthesizing [7]. The FPGA vendors provide their software tools for synthesizing the HDL codes. For synthesizing the HDL design, the libraries and packages used must be supported by these tools [8]. Data type used for HDL designing of DSP algorithm is also an important factor for design synthesis [3]. In this paper, radix-2 based 32-point Fast Fourier Transform algorithm [4] is synthesized to Virtex-5 FPGA. VHDL design of FFT used for synthesis is developed by following two different ways:
1) Using IEEE fixed point package, 'fixed_pkg'[2].
2) Using IEEE package, 'STD_LOGIC_1164' [5].
Data types of DSP algorithm is mostly of 'signed real' type [3]. VHDL provides various fixed point and floating point data types for representing real type data [1]. Since floating point data type is not efficient for synthesizing to FPGA so fixed point data type is used. For implementing DSP algorithm using only IEEE package, 'STD_LOGIC_1164', a method is developed for representing real data type by array of bits. All real arithmetic such as multiplication, addition and subtraction are implemented by specialized algorithms. These algorithms manipulate the bit_vector to implement the real arithmetic.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

II.RADIX-2 ALGORITHM

It is one of the simplest Fast Fourier Transform algorithm in which a Butterfly structure is replicated to get higher order FFT.

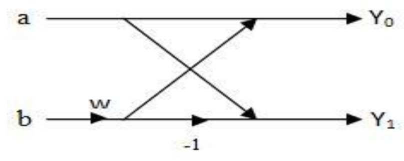


Fig. 1: Butterfly structure

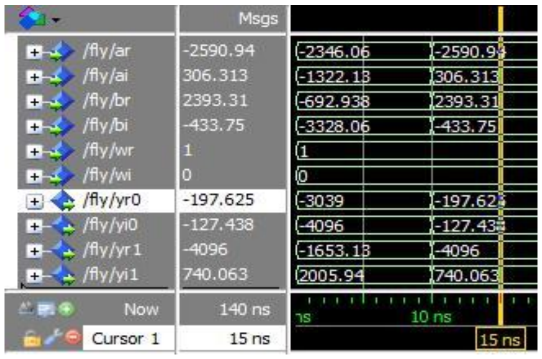
In replicated butterfly structure we change the values of weighted coefficients ‘w’ as per the position of Butterfly. Radix-2 algorithm may be implemented by either decimation in time or decimation in frequency. In decimation in time algorithm we shuffle the order of input while in decimation in frequency algorithm we shuffle the output. Here decimation in time algorithm is used for implementing 32-Point FFT.

III.SIMULATION OF FFT DESIGN USING IEEE FIXED POINT PACKAGE, ‘FIXED_PKG’

IEEE accepted fixed point Package; ‘fixed_pkg’ in VHDL-2008 [1]. This package has powerful operators and functions that specially suits for designing DSP algorithms. 32-Point FFT, radix-2 algorithm is first designed on VHDL using this package then HDL design is comiled and simulated on ModelSim PE Student Edition 10.2a.

A. SIMULATION OF BUTTERFLY STRUCTURE

Here (ar, ai) and (br, bi) are two fixed point real inputs and (yr0 , yi0) and (yr1, yi1) are outputs of Butterfly structure. (wr, wi) is the weighted coefficient of Butterfly structure. Simulation result of Butterfly component is shown in the following figure.



	Msgs		
/fly/ar	-2590.94	-2346.06	-2590.94
/fly/ai	306.313	-1322.13	306.313
/fly/br	2393.31	-692.938	2393.31
/fly/bi	-433.75	-3328.06	-433.75
/fly/wr	1	1	
/fly/wi	0	0	
/fly/yr0	-197.625	-3039	-197.625
/fly/yi0	-127.438	-4096	-127.438
/fly/yr1	-4096	-1653.13	-4096
/fly/yi1	740.063	2005.94	740.063
Now	140 ns	10 ns	
Cursor 1	15 ns	15 ns	

Fig. 2: Simulation result of Butterfly

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

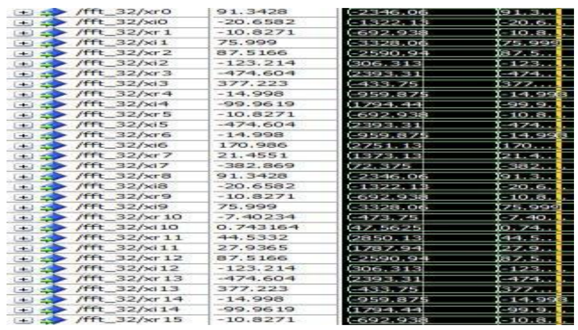
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

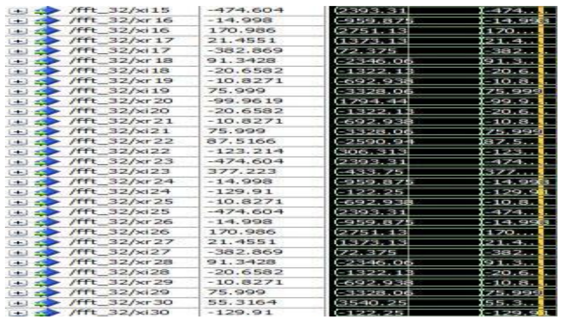
Vol. 2, Issue 8, August 2013

B. SIMULATION OF 32-POINT FFT

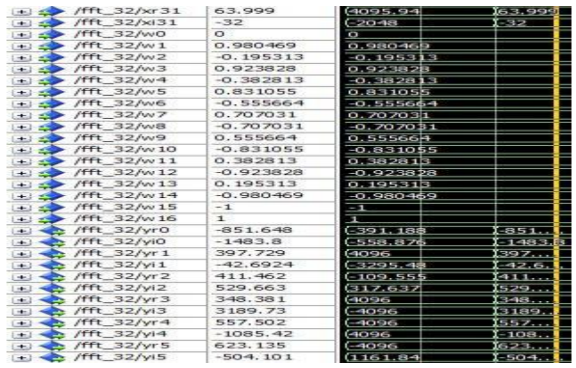
In this 32- Point FFT design, above simulated Butterfly is used as component. Butterfly component is used in five stages and in each stage sixteen instances of Butterfly are used. Here xr and xi are real and imaginary parts of input x. Similarly yr, yi and wr, wi are real and imaginary parts of y and w respectively. There are 32 complex inputs, 32 complex outputs and 17 different weighted coefficients. Simulation result of 32-Point FFT is shown in the following figure.



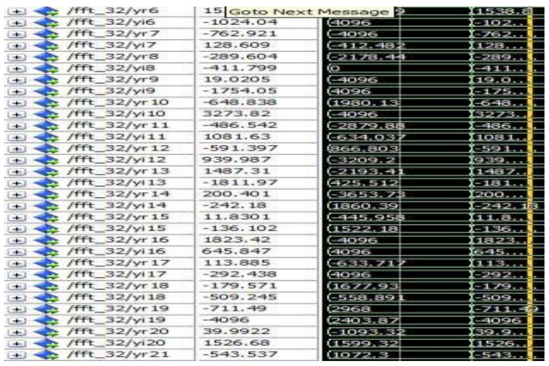
```
/fft_32/xr0     91.3428      -2346.06   91.3...
/fft_32/xi0     -20.6582     -1322.13   -20.6...
/fft_32/xr1     -10.8271     -692.938   -10.8...
/fft_32/xi1     75.999       -3328.06   75.999
/fft_32/xr2     87.5166      -2590.94   87.5...
/fft_32/xi2     -123.214     306.313    -123...
/fft_32/xr3     -474.604     2393.31    -474...
/fft_32/xi3     377.223      -433.75    377...
/fft_32/xr4     -14.998      -959.875   -14.998
/fft_32/xi4     -99.9619     1794.44    -99.9...
/fft_32/xr5     -10.8271     -692.938   -10.8...
/fft_32/xi5     -474.604     2393.31    -474...
/fft_32/xr6     -14.998      -959.875   -14.998
/fft_32/xi6     170.986      2751.13    170...
/fft_32/xr7     21.4551      1373.13    21.4...
/fft_32/xi7     -382.869     72.375     -382...
/fft_32/xr8     91.3428      -2346.06   91.3...
/fft_32/xi8     -20.6582     -1322.13   -20.6...
/fft_32/xr9     -10.8271     -692.938   -10.8...
/fft_32/xi9     75.999       -3328.06   75.999
/fft_32/xr10    -7.40234     -473.75    -7.40...
/fft_32/xi10    0.743164     47.5625    0.74...
/fft_32/xr11    44.5332      2850.13    44.5...
/fft_32/xi11    27.9365      1787.94    27.9...
/fft_32/xr12    87.5166      -2590.94   87.5...
/fft_32/xi12    -123.214     306.313    -123...
/fft_32/xr13    -474.604     2393.31    -474...
/fft_32/xi13    377.223      -433.75    377...
/fft_32/xr14    -14.998      -959.875   -14.998
/fft_32/xi14    -99.9619     1794.44    -99.9...
/fft_32/xr15    -10.8271     -692.938   -10.8...
```



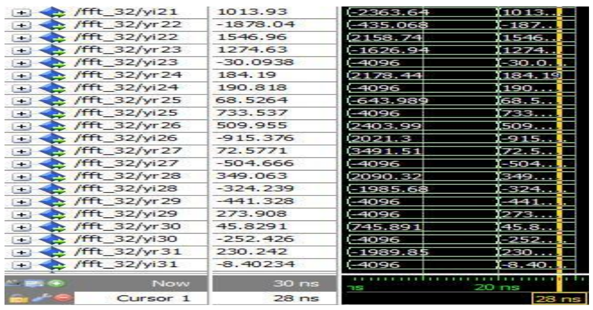
```
/fft_32/xi15    -474.604     2393.31    -474...
/fft_32/xr16    -14.998      -959.875   -14.998
/fft_32/xi16    170.986      2751.13    170...
/fft_32/xr17    21.4551      1373.13    21.4...
/fft_32/xi17    -382.869     72.375     -382...
/fft_32/xr18    91.3428      -2346.06   91.3...
/fft_32/xi18    -20.6582     -1322.13   -20.6...
/fft_32/xr19    -10.8271     -692.938   -10.8...
/fft_32/xi19    75.999       -3328.06   75.999
/fft_32/xr20    -99.9619     1794.44    -99.9...
/fft_32/xi20    -20.6582     -1322.13   -20.6...
/fft_32/xr21    -10.8271     -692.938   -10.8...
/fft_32/xi21    75.999       -3328.06   75.999
/fft_32/xr22    87.5166      -2590.94   87.5...
/fft_32/xi22    -123.214     306.313    -123...
/fft_32/xr23    -474.604     2393.31    -474...
/fft_32/xi23    377.223      -433.75    377...
/fft_32/xr24    -14.998      -959.875   -14.998
/fft_32/xi24    -129.91      -122.25    -129.91
/fft_32/xr25    -10.8271     -692.938   -10.8...
/fft_32/xi25    -474.604     2393.31    -474...
/fft_32/xr26    -14.998      -959.875   -14.998
/fft_32/xi26    170.986      2751.13    170...
/fft_32/xr27    21.4551      1373.13    21.4...
/fft_32/xi27    -382.869     72.375     -382...
/fft_32/xr28    91.3428      -2346.06   91.3...
/fft_32/xi28    -20.6582     -1322.13   -20.6...
/fft_32/xr29    -10.8271     -692.938   -10.8...
/fft_32/xi29    75.999       -3328.06   75.999
/fft_32/xr30    55.3164      3540.25    55.3...
/fft_32/xi30    -129.91      -122.25    -129.91
```



```
/fft_32/xr31    63.999       4095.94    63.999
/fft_32/xi31    -32          -2048      -32
/fft_32/w0      0            0
/fft_32/w1      0.980469     0.980469
/fft_32/w2      -0.195313    -0.195313
/fft_32/w3      0.923828     0.923828
/fft_32/w4      -0.382813    -0.382813
/fft_32/w5      0.831055     0.831055
/fft_32/w6      -0.555664    -0.555664
/fft_32/w7      0.707031     0.707031
/fft_32/w8      -0.707031    -0.707031
/fft_32/w9      0.555664     0.555664
/fft_32/w10     -0.831055    -0.831055
/fft_32/w11     0.382813     0.382813
/fft_32/w12     -0.923828    -0.923828
/fft_32/w13     0.195313     0.195313
/fft_32/w14     -0.980469    -0.980469
/fft_32/w15     -1           -1
/fft_32/w16     1            1
/fft_32/yr0     -851.648     -391.188   -851...
/fft_32/yi0     -1483.8      -558.876   -1483.8
/fft_32/yr1     397.729      4096       397...
/fft_32/yi1     -42.6924     -3295.48   -42.6...
/fft_32/yr2     411.462      -109.555   411...
/fft_32/yi2     529.663      317.637    529...
/fft_32/yr3     348.381      4096       348...
/fft_32/yi3     3189.73      -4096      3189...
/fft_32/yr4     557.502      -4096      557...
/fft_32/yi4     -1085.42     4096       -108...
/fft_32/yr5     623.135      -4096      623...
/fft_32/yi5     -504.101     1161.84    -504...
```



```
/fft_32/yr6     15 Goto Next Message 9  1538.8
/fft_32/yi6     -1024.04     4096       -102...
/fft_32/yr7     -762.921     -4096      -762...
/fft_32/yi7     128.609      -412.482   128...
/fft_32/yr8     -289.604     -2178.44   -289...
/fft_32/yi8     -411.799     0          -411...
/fft_32/yr9     19.0205      -4096      19.0...
/fft_32/yi9     -1754.05     4096       -175...
/fft_32/yr10    -648.838     1980.13    -648...
/fft_32/yi10    3273.82      -4096      3273...
/fft_32/yr11    -486.542     -2879.88   -486...
/fft_32/yi11    1081.63      -634.037   1081...
/fft_32/yr12    -591.397     866.803    -591...
/fft_32/yi12    939.987      -3209.2    939...
/fft_32/yr13    1487.31      -2193.41   1487...
/fft_32/yi13    -1811.97     425.512    -181...
/fft_32/yr14    200.401      -3653.73   200...
/fft_32/yi14    -242.18      1860.39    -242.18
/fft_32/yr15    11.8301      -445.958   11.8...
/fft_32/yi15    -136.102     1522.18    -136...
/fft_32/yr16    1823.42      -4096      1823...
/fft_32/yi16    645.847      4096       645...
/fft_32/yr17    113.885      -633.717   113...
/fft_32/yi17    -292.438     4096       -292...
/fft_32/yr18    -179.571     1677.93    -179...
/fft_32/yi18    -509.245     -558.891   -509...
/fft_32/yr19    -711.49      2968       -711.49
/fft_32/yi19    -4096        2403.87    -4096
/fft_32/yr20    39.9922      -1093.32   39.9...
/fft_32/yi20    1526.68      1599.32    1526...
/fft_32/yr21    -543.537     1072.3     -543...
```



```
/fft_32/yi21    1013.93      -2363.64   1013...
/fft_32/yr22    -1878.04     -435.068   -187...
/fft_32/yi22    1546.96      2158.74    1546...
/fft_32/yr23    1274.63      -1626.94   1274...
/fft_32/yi23    -30.0938     -4096      -30.0...
/fft_32/yr24    184.19       2178.44    184.19
/fft_32/yi24    190.818      -4096      190...
/fft_32/yr25    68.5264      -643.989   68.5...
/fft_32/yi25    733.537      -4096      733...
/fft_32/yr26    509.955      2403.99    509...
/fft_32/yi26    -915.376     2021.3     -915...
/fft_32/yr27    72.5771      3491.51    72.5...
/fft_32/yi27    -504.666     -4096      -504...
/fft_32/yr28    349.063      2090.32    349...
/fft_32/yi28    -324.239     -1985.68   -324...
/fft_32/yr29    -441.328     -4096      -441...
/fft_32/yi29    273.908      -4096      273...
/fft_32/yr30    45.8291      745.891    45.8...
/fft_32/yi30    -252.426     -4096      -252...
/fft_32/yr31    230.242      -1989.85   230...
/fft_32/yi31    -8.40234     -4096      -8.40...
Now             30 ns        ns         20 ns
Cursor 1        28 ns                   28 ns
```

Fig. 3: Simulated result of 32-point FFT

C. ISSUES ON SYNTHESIZING THIS DESIGN

Fixed point package, fixed_pkg is brought to VHDL-2008 by IEEE. It is said that the designs using this package will be synthesizable. All the data structures used in this package are fixed point.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

This package contains powerful operators and functions which makes it very efficient in designing DSP algorithm. But, still this package is not absolute synthesizable because it is not fully supported by synthesizer tools.

IV.IMPLEMENTATION OF FFT USING IEEE PACKAGE, 'STD_LOGIC_1164' ALONE

For implementing the DSP algorithms using IEEE package, 'STD_LOGIC_1164' alone, a data type is required which may implement both the real and complex numbers. Its real data type is of floating point, so it will not be synthesizable. So a method is developed which represent the real and complex numbers by 'bit_vector'.

A. *REPRESENTATION OF REAL AND COMPLEX NUMBERS BY 'BIT_VECTOR'*

'A' is an array of bits of length 'L' in which M bits represents fraction part and N bits represents whole part.
$A=b_{(N-1)}\ldots\ldots b_2b_1b_0\ b_1\ldots\ldots b_{(M-1)}b_M$
The numbers N and M are chosen depending on the precision and range of real number.
Here important point is that the software and FPGA will treat this array of bits 'A' as simply a bit vector of length 'M+N'.

B. *ALGORITHMS FOR ARITHMETIC OPERATION ON THIS DATA STRUCTURE*

Specialized algorithms are developed which will operate on above defined data structure and manipulate the array of bits in such a way that they will fulfill the functions of complex and real numbers. For the 32-Point FFT calculation, addition, subtraction and multiplication algorithms are used. Multiplier algorithm is explained in following context.

C. *ALGORITHM FOR MULTIPLIER*

This multiplier is designed for multiplication operation in FFT design. It takes two bit vectors as input and gives a bit vector as output with its length equal to multiplicand.

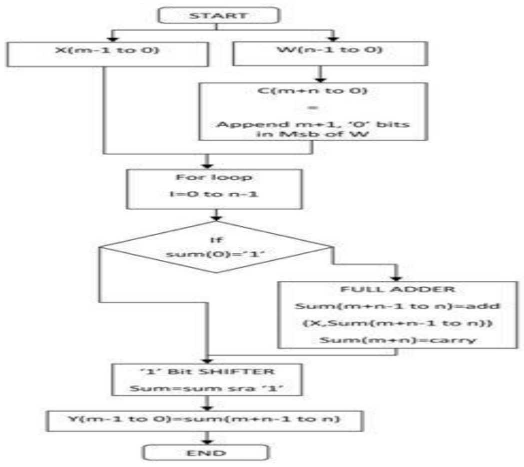


Fig. 4: Flow Chart for Multiplier Algorithm

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

D. SYNTHESIS AND SIMULATION OF FFT USING IEEE PACKAGE, 'STD_LOGIC_1164'

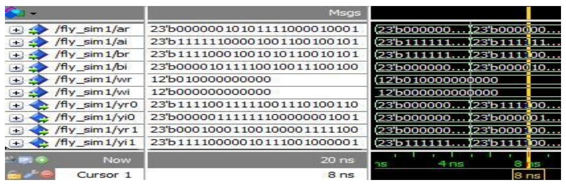
These results are compiled and simulated on Modelsim PE Student Edition 10.2a and synthesized on Xilinx ISE 10.1 design suite. For design simulation 'Xilinx Virtex-5' FPGA is used.

Table I: Target FPGA Properties

Target FPGA Properties	
Family	Virtex5
Device	XC5VLX30
Package	FF324
Speed	-3

1. SIMULATION OF BUTTERFLY STRUCTURE

Simulated result is shown in the following figure. As it was shown in figure-1, it has two complex inputs, two complex outputs and one complex weighted coefficient. Each input and output is of 23 bits length and coefficient is of 12 bits length. Least significant 10 bits are fraction bits.



```
                     Msgs
/fly_sim1/ar    23'b00000010101111000010001     23'b000000...  23'b000000...
/fly_sim1/ai    23'b11111100001001100100101     23'b111111...  23'b111111...
/fly_sim1/br    23'b11110001001010110010101     23'b111111...  23'b111100...
/fly_sim1/bi    23'b00001011110010011100100     23'b000000...  23'b000010...
/fly_sim1/wr    12'b010000000000                12'b010000000000
/fly_sim1/wi    12'b000000000000                12'b000000000000
/fly_sim1/yr0   23'b11110011111001110100110     23'b000000...  23'b111100...
/fly_sim1/yi0   23'b00000111111100000001001     23'b000000...  23'b000001...
/fly_sim1/yr1   23'b00010001100100001111100     23'b000000...  23'b000100...
/fly_sim1/yi1   23'b11110000010111001000001     23'b111111...  23'b111100...
Now             20 ns                           ns    4 ns    8 ns
Cursor 1        8 ns                                          8 ns
```

Fig. 5: simulated result of Butterfly component

2. SYNTHESIS RESULTS OF BUTTERFLY STRUCTURE

The HDL design of Butterfly is synthesized with Xilinx ISE 10.1 design suite. RTL view of Butterfly structure is shown in the following figure. This RTL structure is automatically generated after synthesizing the design with Xilinx design suite. The next is synthesis design summary of Butterfly on Virtex-5 FPGA.

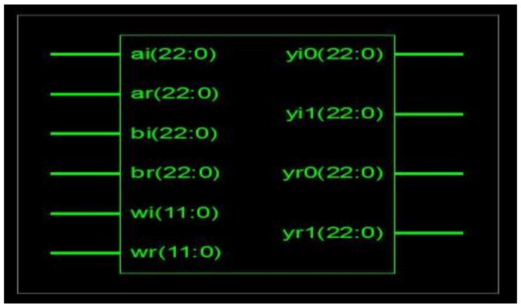
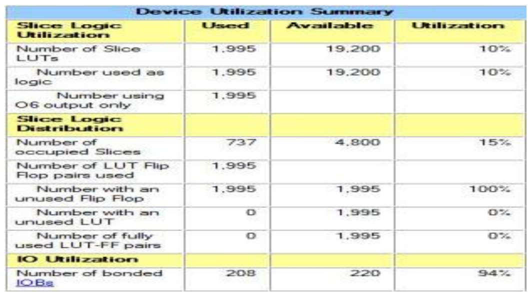


Fig. 6: RTL view of Butterfly component



Device Utilization Summary			
Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	1,995	19,200	10%
Number used as logic	1,995	19,200	10%
Number using O6 output only	1,995		
Slice Logic Distribution			
Number of occupied Slices	737	4,800	15%
Number of LUT Flip Flop pairs used	1,995		
Number with an unused Flip Flop	1,995	1,995	100%
Number with an unused LUT	0	1,995	0%
Number of fully used LUT-FF pairs	0	1,995	0%
IO Utilization			
Number of bonded IOBs	208	220	94%

Fig. 7: Synthesis design summary of Butterfly component

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

3. TIMING SUMMARY

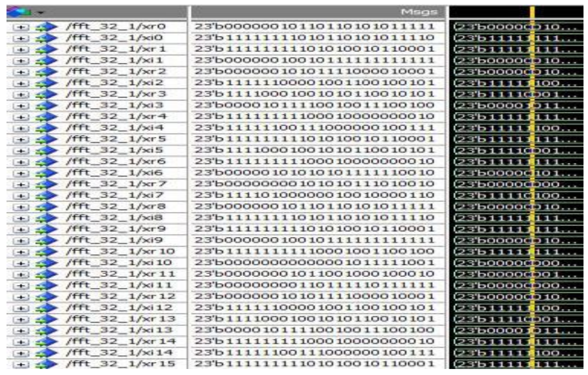
Maximum combinational path delay of designed FFT-32 on FPGA is 95.814ns.

Table II: Time delay of Butterfly component

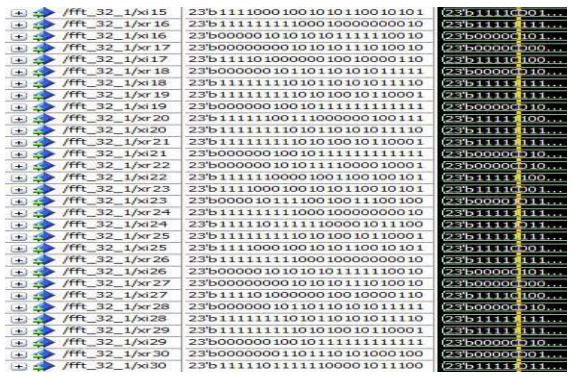
Delay type	Delay (ns)	Delay (%)
Logic	15.522	16.0
Route	81.292	84.0
Total	95.814	100.0

4. SIMULATION OF 32-POINT FFT

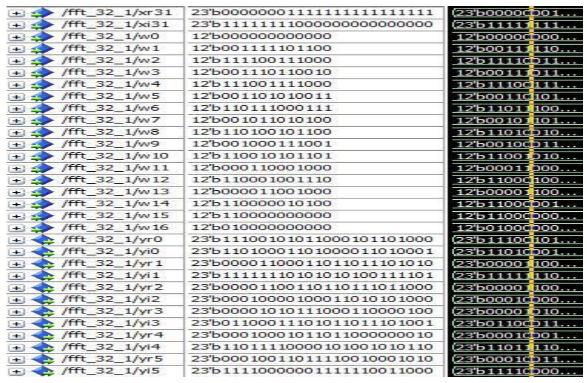
In the 32-Point FFT design, above described Butterfly design is used as component. Total 80 instances of Butterfly are used in this design.



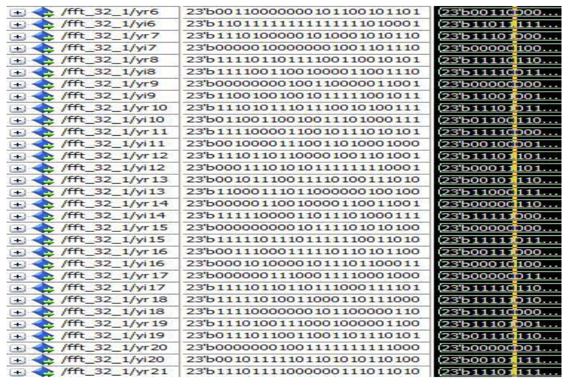
```
                     Msgs
/fft_32_1/xr0    23'b00000010110110101011111
/fft_32_1/xi0    23'b11111111010110101011110
/fft_32_1/xr1    23'b111111111101010010110001
/fft_32_1/xi1    23'b00000010010111111111111
/fft_32_1/xr2    23'b00000010101111000010001
/fft_32_1/xi2    23'b11111100001001100100101
/fft_32_1/xr3    23'b11110001001010110010101
/fft_32_1/xi3    23'b00001011110010011100100
/fft_32_1/xr4    23'b11111111100010000000010
/fft_32_1/xi4    23'b11111100111000000100111
/fft_32_1/xr5    23'b111111111101010010110001
/fft_32_1/xi5    23'b11110001001010110010101
/fft_32_1/xr6    23'b11111111100010000000010
/fft_32_1/xi6    23'b00000101010101111110010
/fft_32_1/xr7    23'b00000000101010111010010
/fft_32_1/xi7    23'b11110100000010010000110
/fft_32_1/xr8    23'b00000010110110101011111
/fft_32_1/xi8    23'b11111111010110101011110
/fft_32_1/xr9    23'b111111111101010010110001
/fft_32_1/xi9    23'b00000010010111111111111
/fft_32_1/xr10   23'b111111111110001001100100
/fft_32_1/xi10   23'b000000000000001011111001
/fft_32_1/xr11   23'b00000001011001000100010
/fft_32_1/xi11   23'b00000000110111110111111
/fft_32_1/xr12   23'b00000010101111000010001
/fft_32_1/xi12   23'b11111100001001100100101
/fft_32_1/xr13   23'b11110001001010110010101
/fft_32_1/xi13   23'b00001011110010011100100
/fft_32_1/xr14   23'b11111111100010000000010
/fft_32_1/xi14   23'b11111100111000000100111
/fft_32_1/xr15   23'b111111111101010010110001
```



```
/fft_32_1/xi15   23'b11110001001010110010101
/fft_32_1/xr16   23'b11111111100010000000010
/fft_32_1/xi16   23'b00000101010101111110010
/fft_32_1/xr17   23'b00000000101010111010010
/fft_32_1/xi17   23'b11110100000010010000110
/fft_32_1/xr18   23'b00000010110110101011111
/fft_32_1/xi18   23'b11111111010110101011110
/fft_32_1/xr19   23'b111111111101010010110001
/fft_32_1/xi19   23'b00000010010111111111111
/fft_32_1/xr20   23'b11111100111000000100111
/fft_32_1/xi20   23'b11111111010110101011110
/fft_32_1/xr21   23'b111111111101010010110001
/fft_32_1/xi21   23'b00000010010111111111111
/fft_32_1/xr22   23'b00000010101111000010001
/fft_32_1/xi22   23'b11111100001001100100101
/fft_32_1/xr23   23'b11110001001010110010101
/fft_32_1/xi23   23'b00001011110010011100100
/fft_32_1/xr24   23'b11111111100010000000010
/fft_32_1/xi24   23'b11111011111100001011100
/fft_32_1/xr25   23'b111111111101010010110001
/fft_32_1/xi25   23'b11110001001010110010101
/fft_32_1/xr26   23'b11111111100010000000010
/fft_32_1/xi26   23'b00000101010101111110010
/fft_32_1/xr27   23'b00000000101010111010010
/fft_32_1/xi27   23'b11110100000010010000110
/fft_32_1/xr28   23'b00000010110110101011111
/fft_32_1/xi28   23'b11111111010110101011110
/fft_32_1/xr29   23'b111111111101010010110001
/fft_32_1/xi29   23'b00000010010111111111111
/fft_32_1/xr30   23'b00000001101110101000100
/fft_32_1/xi30   23'b11111011111100001011100
```



```
/fft_32_1/xr31   23'b00000001111111111111111
/fft_32_1/xi31   23'b11111111000000000000000
/fft_32_1/w0     12'b000000000000
/fft_32_1/w1     12'b001111101100
/fft_32_1/w2     12'b111100111000
/fft_32_1/w3     12'b001110110010
/fft_32_1/w4     12'b111001111000
/fft_32_1/w5     12'b001101010011
/fft_32_1/w6     12'b110111000111
/fft_32_1/w7     12'b001011010100
/fft_32_1/w8     12'b110100101100
/fft_32_1/w9     12'b001000111001
/fft_32_1/w10    12'b110010101101
/fft_32_1/w11    12'b000110001000
/fft_32_1/w12    12'b110001001110
/fft_32_1/w13    12'b000011001000
/fft_32_1/w14    12'b110000010100
/fft_32_1/w15    12'b110000000000
/fft_32_1/w16    12'b010000000000
/fft_32_1/yr0    23'b11100101011000101101000
/fft_32_1/yi0    23'b11010001101000011010001
/fft_32_1/yr1    23'b00001100011011011101010
/fft_32_1/yi1    23'b111111110101010100111101
/fft_32_1/yr2    23'b00001100110110111011000
/fft_32_1/yi2    23'b00010000100011010101000
/fft_32_1/yr3    23'b00001010111000110000100
/fft_32_1/yi3    23'b01100011101011011101001
/fft_32_1/yr4    23'b00010001011011000000010
/fft_32_1/yi4    23'b11011110000101001010110
/fft_32_1/yr5    23'b00010011011110010001010
/fft_32_1/yi5    23'b11110000000111110011000
```



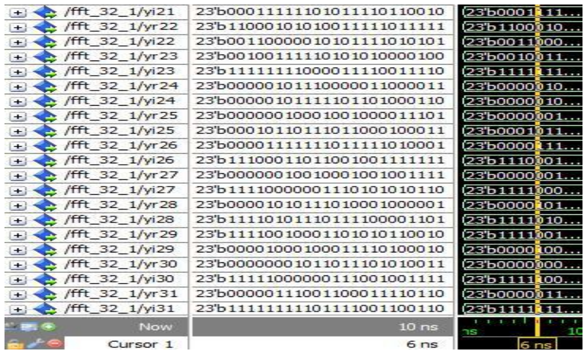
```
/fft_32_1/yr6    23'b00110000000101100101101
/fft_32_1/yi6    23'b1101111111111111111010001
/fft_32_1/yr7    23'b11101000001010001010110
/fft_32_1/yi7    23'b00000100000001001101110
/fft_32_1/yr8    23'b1111011011100110010101
/fft_32_1/yi8    23'b11110011001000011001110
/fft_32_1/yr9    23'b00000000100110000011001
/fft_32_1/yi9    23'b11001001001011111001011
/fft_32_1/yr10   23'b11101011101110010100111
/fft_32_1/yi10   23'b01100110010011101000111
/fft_32_1/yr11   23'b11110000110010111010101
/fft_32_1/yi11   23'b00100001110011010001000
/fft_32_1/yr12   23'b11101101100001001101001
/fft_32_1/yi12   23'b00011101010111111110001
/fft_32_1/yr13   23'b00101110011110100111010
/fft_32_1/yi13   23'b11000111011000000100100
/fft_32_1/yr14   23'b00000110010000110011001
/fft_32_1/yi14   23'b11111000011011101000111
/fft_32_1/yr15   23'b00000000010111101010100
/fft_32_1/yi15   23'b11111011101111110011010
/fft_32_1/yr16   23'b00111000111110110101100
/fft_32_1/yi16   23'b00010100001011101100011
/fft_32_1/yr17   23'b00000011100011110001000
/fft_32_1/yi17   23'b11110110110111000111101
/fft_32_1/yr18   23'b11111010011000110111000
/fft_32_1/yi18   23'b11110000000101100000110
/fft_32_1/yr19   23'b11101001110001000001100
/fft_32_1/yi19   23'b01110110011001101110101
/fft_32_1/yr20   23'b00000001001111111111000
/fft_32_1/yi20   23'b00101111101101010110100
/fft_32_1/yr21   23'b11101111000000111011010
```

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013



```
/fft_32_1/yi21   23'b00011111101011110110010   23'b0001111...
/fft_32_1/yr22   23'b11000101010011111011111   23'b1100010...
/fft_32_1/yi22   23'b00110000010101111010101   23'b0011000...
/fft_32_1/yr23   23'b00100111110101010000100   23'b0010011...
/fft_32_1/yi23   23'b11111111000011110011110   23'b1111111...
/fft_32_1/yr24   23'b00000101110000011000011   23'b0000010...
/fft_32_1/yi24   23'b00000101111101101000110   23'b0000010...
/fft_32_1/yr25   23'b00000010001001000011101   23'b0000001...
/fft_32_1/yi25   23'b00010110111011000100011   23'b0001011...
/fft_32_1/yr26   23'b00001111111011111010001   23'b0000111...
/fft_32_1/yi26   23'b11100011011001001111111   23'b1110001...
/fft_32_1/yr27   23'b00000010010001001001111   23'b0000001...
/fft_32_1/yi27   23'b11110000000111010101010110   23'b1111000...
/fft_32_1/yr28   23'b00001010111010001000001   23'b0000101...
/fft_32_1/yi28   23'b11110101110111100001101   23'b1111010...
/fft_32_1/yr29   23'b11110010001101010110010   23'b1111001...
/fft_32_1/yi29   23'b00001000100011110100010   23'b0000100...
/fft_32_1/yr30   23'b00000001011011101010011   23'b0000000...
/fft_32_1/yi30   23'b11111000000111001001111   23'b1111100...
/fft_32_1/yr31   23'b00000111001100011110110   23'b0000011...
/fft_32_1/yi31   23'b11111111101111001100110   23'b1111111...
            Now                          10 ns
       Cursor 1                           6 ns   6 ns
```

Fig. 8: Simulated result of 32-Point FFT

5. SYNTHESIS OF 32-POINT FFT

The proposed design of 32-FFT block is synthesized using the Xilinx ISE 10.1 design suite. Thus the RTL block obtained after synthesizing the design is shown below.

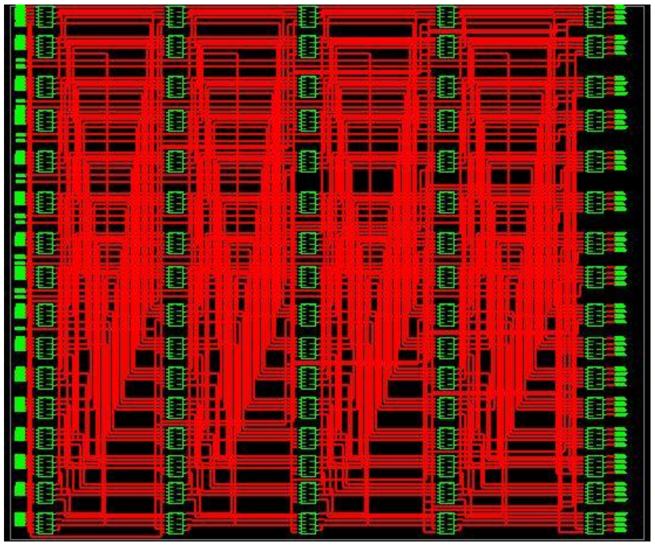


Fig. 9: Synthesized Internal RTL architecture of 32-Point FFT

From the above synthesized Internal RTL architecture it clear that whole architecture is divided in five stages and each stage comprises sixteen instances of Butterfly, thus total eighty instances of Butterfly are visible in above shown RTL architecture.

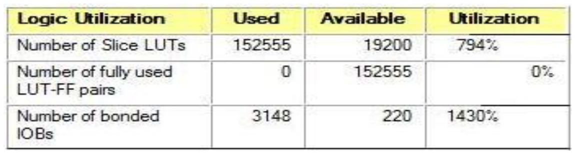
The next is synthesis design summary of 32-Point FFT on Virtex-5 FPGA, it shows the features of the Virtex-5 FPGA used by Xilinx design suite for proposed work. Maximum combinational path delay: 466.732ns.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013



Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	152555	19200	794%
Number of fully used LUT-FF pairs	0	152555	0%
Number of bonded IOBs	3148	220	1430%

Fig. 10: Synthesis design summary of 32-Point FFT

Delay type	Delay (ns)	Delay (%)
Logic	65.314	14.0
Route	401.418	86.0
Total	466.732	100.0

Table III: Time delay of 32-Point FFT

V.COMPARISON OF RESULT WITH MATLAB 7.11.0.584

Simulated results of synthesizable design of 32-Point FFT using 'STD_logic_1164' package are compared with Matlab and the percentage error is calculated for each output. The comparison table is shown below.

Table IV: Comparison of 32-Point FFT results with Matlab

FFT Input			Design Output				Matlab Output		%Error	
			Binary(10 fraction bits)		Decimal equivalent					
s/n	Xr	Xi	Yr	Yi	Yr	Yi	Yr	Yi	Er	Ei
0	91.3428	-20.6582	1110010101100.0101101000	1101000110100.0011010001	-851.6484	-1483.7958	-851.64613	-1483.7941	0.00027	0.00011
1	-10.8271	75.999	0000110001101.1101110101	1111111010101.0100111101	397.7285	-42.69042	397.66331	-42.71583	0.01639	0.05948
2	87.5166	-123.214	0000110011011.0111011000	0001000010001.1010101000	411.4609	529.66406	411.52641	529.69893	0.01591	0.00658
3	-474.604	377.223	0000101011100.0110000100	0110001110101.1011101001	348.3789	3189.72753	347.59773	3191.11167	0.22473	0.04337
4	-14.998	-99.9619	0001000101101.1000000010	1101111000010.1001010110	557.5019	-1085.4160	557.52852	1085.55534	0.00476	0.01283
5	-10.8271	-474.604	0001001101111.0010001010	1111000000111.1110011000	623.1347	-504.10156	623.08508	-504.56483	0.00793	0.09181
6	-14.998	170.986	0011000000010.1100101101	1101111111111.1111010001	1528.793	-1024.0458	1538.68797	-1024.3272	0.64301	0.02746
7	21.4551	-382.869	1110100000101.0001010110	0000010000000.1001101110	-762.9160	128.60742	-763.26689	129.02958	0.04597	0.32718
8	91.3428	-20.6582	1111011011110.0110010101	1111001100100.0011001110	-289.6044	-411.79882	-289.60446	-411.79806	0.00001	0.00018
9	-10.8271	75.999	0000000010011.1000011001	1100100100101.1111001011	19.52441	-1754.0517	19.40969	-1754.5238	0.59104	0.02690
10	-7.40234	0.743164	1110101110111.0010100111	0110011001001.1101000111	-648.8369	3273.81933	-649.19492	3273.92517	0.05514	0.00323
11	44.5332	27.9365	1111000011001.0111010101	0010000111001.1010001000	-486.5419	1081.63281	-486.75619	1082.09167	0.04005	0.04240
12	87.5166	-123.214	1110110110000.1001101001	0001110101011.1111111001	-591.3974	939.98535	-591.44750	940.05443	0.00846	0.00734
13	-474.604	377.223	0010111001111.0100111010	1100011101100.0000100100	1487.306	-1811.9648	1487.49784	-1812.4950	0.01285	0.02925
14	-14.998	-99.9619	0000011001000.0110011001	1111100001101.1101000111	200.3994	-242.18066	200.49120	-242.12677	0.04578	0.02258
15	-10.8271	-474.604	0000000001011.1001010100	1111101110111.1110011010	11.58203	-136.09960	11.51392	-136.05351	0.59154	0.03387
16	-14.998	170.986	0011100011111.0110101100	0001010000101.1101100011	1823.417	645.84667	1823.41646	645.84286	0.00008	0.00059
17	21.4551	-382.869	0000001110001.1110001000	1111011011011.1000111101	113.8828	-292.44042	113.95275	-292.35080	0.06137	0.03065
18	91.3428	-20.6582	1111101001100.0110111000	1111000000010.1100000110	-179.570	-509.24414	-179.61587	-509.40605	0.02536	0.03178
19	-10.8271	75.999	1110100111000.1000001100	0111011001100.1101110101	-711.488	* 3788.86426	-710.86521	* - 4404.59939	0.08765	*
20	-99.9619	-20.6582i	0000000100111.1111111000	0010111110110.1010110100	39.99218	1526.67578	39.96539	1526.81502	0.06703	0.00912
21	-10.8271	75.999	1110111100000.0111011010	0001111110101.1110110010	-543.537	1013.92382	-543.65921	1014.39614	0.02246	0.04656
22	87.5166	-123.214	1100010101001.1111011111	0011000001010.1111010101	-1878.03	1546.95800	-1877.9584	1547.29390	0.00393	0.02170
23	-474.604	377.223	0010011111010.1010000100	1111111100001.0110011110	1274.628	-30.59570	1275.09324	-30.50869	0.03641	0.28519
24	-14.998	-129.91	0000010111110.1101000110	0000010111110.1101000110	184.1904	190.81835	184.19054	190.81853	0.00006	0.00009
25	-10.8271	-474.604	0000001000100.1000011101	0001011011101.1000100011	68.52832	733.53417	68.10462	733.92532	0.62213	0.05329
26	-14.998	170.986	0000111111101.1111010001	1110001101100.1001111111	509.9541	-915.37597	510.29278	-915.35686	0.06637	0.00208
27	21.4551	-382.869	0000001001000.1001001111	1111000000111.0101010110	72.57714	-504.66601	72.86700	-505.08154	0.39779	0.08227
28	91.3428	-20.6582	0000101011101.0001000001	1111010111011.1100001101	349.0634	-324.23730	349.11397	-324.30651	0.01446	0.02134
29	-10.8271	75.999	1111001000110.1010110010	0000100010001.1110100010	-441.326	273.90820	-441.31329	274.44890	0.00291	0.19701
30	55.3164	-129.91	0000000101101.1101010011	1111100000011.1001001111	45.83105	-252.42285	45.77084	-252.53310	0.13154	0.04365
31	63.999	-32.00	0000011100110.0011110110	0000011100110.0011110110	230.2402	-8.40039	230.52839	-8.42202	0.12500	0.25682
Average Percentage Error-									0.149 0%	0.055 9%
									0.10247%	

**4404.59939 is outside the vector range (4095.99)*

VI. CONCLUSION

The HDL design of 32-Point FFT is implemented with IEEE fixed point package, 'fixed_pkg'. This design works properly on Modelsim but is non-synthesizable. A new data structure is devised which is actually a bit vector that fulfills all the complex and real data structure needs. This data structure is completely defined by IEEE package,'STD_logic_1164. For implementing DSP algorithms using this data structure some specialized arithmetic algorithms are designed. 32-Point FFT

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2013

is again implemented with this data structure and this design is absolutely synthesizable. Simulated results of synthesizable 32-Point FFT design are compared with Matlab results, and we get average error of 0.1%. This shows that this method of designing and implementing DSP algorithms is very efficient and completely synthesizable.

REFERENCES

[1] 1076-2008- IEEE Standard VHDL Language Reference Manual, DOI: 10.1109/IEEESTD.2009.4772740, 2009.
[2] David Bishop, fixed Point Package User's Guide. URL: www.vhdl.org/fphdl/Fixed_ug.pdf.
[3] Uwe Meyer-Base, Digital Signal Processing With Field Programmable Gate Arrays 3E, 2007
[4] John G. Proakis, Digital Signal Processing: Principles, algorithms, And Applications, 4E, 2007.
[5] Charles H. Roth, Digital Systems Design using VHDL, 1998.
[6] Douglas J. Smith: A practical Guide For Designing, Synthesizing And Simulating Asics And FPGAs Using VHDL Or Verilog, 1996.
[7] Synthesis Forum of Xilinx, UTL: http://forums.xilinx.com/t5/Synthesis/compilation-of-ieee-proposed-library- fails-ISE-13-2/td-p/200101.
[8] Software Manual of Xilinx ISE Design suite.