

# A NEW TECHNIQUE FOR IMPROVING THE POWER QUALITY IN POWER TRANSFORMERS BY FPGA

**[1]P.RAM KISHORE KUMAR REDDY, [2]DR.K.S.R.ANJANEYULU**
[1]Asst.Prof., Department of Electrical Engineering, M.G.I.T., Hyderabad, India – 500075.
[2]Prof., Department of Electrical Engineering, JNTUCE, Anantapur, India – 514002.
Email : prkkreddy@rediffmail.com , ksralu@yahoo.co.uk

**ABSTRACT**

Maintenance of good quality of power supply is essential for reliability and stability of power system. The effects due to power transformer switching and the transient effects are investigated in this paper. The DWT based techniques are suggested in the past to analyze the transient effects and to respond at a faster rate to improve the quality of power system and protection. These measures were though efficient in providing better solution but found to be high resource consuming under remote applications. This high resource consumption results in slower response and reduces the reliability issue in power system. A FPGA based DWT architecture is suggested to improve the efficiency of estimation and response in the power system. The Field Programmable Gate Arrays (FPGA) based DWT architecture is evaluated with theoretical results from MATLB and were observed to be meeting the accuracy of estimation.

**Key words** : *Power Quality, Wavelet Transform, Power Transformer, Digital Modeling, VHDL, FPGA Implementation*.

## 1. INTRODUCTION

The electric power requirement is increasing due to increase in demand from electrical utilities. Since power system is AC in nature, the power transformer is commanded as one of the most important equipments in power system. Detecting minor faults in power transformer has become one of the most important requirements for extending the power quality of the power system. In recent years power quality is one of the primary concerns of the utilities, since lack of quality in power may cause malfunctions, instability, short lifetime, and so on. In past ten years it is observed that the most important causes which take the responsibility for the power system failures and transformer damages are the transformer winding deformations.

Therefore to safe guard the quality of power it is required to check whether the strength of the insulation of the winding can withstand for severe faults. The withstanding capability of the insulation can be checked by impulse test. The standard method of impulse testing of high voltage power transformer is associated with the problems regarding identification of minute failures particularly inter- turn faults. The[(1)] conventional method of impulse testing of transformer is based on the comparison of the applied voltage and the neutral current (oscillograms take at reduced and full voltage). A minor difference between the compared oscillogram can be inductive of inter-turn failure, which disqualifies a large and expensive transformer. But impulse test could not detect minor faults since high voltage impulse generator produces a slightly different impulse waveform at the full and reduced levels. This inturn will cause a difference between the compared neutral current oscillogram, which, according to the existing standards may be interpreted as a transformer fault. Another drawback of the recent test is the rather crude evaluation of the chopped impulse test. Actually this is the most critical test for the HV terminal section of the winding, because of the steepness and amplitude of the applied voltage. Neutral current comparison is not applicable here since the time to chop cannot be controlled. Consequently successive oscillogram of the neutral current may show a considerable difference due to the scatter in the chopped impulse duration.

In this aspect, the wavelet analysis has gained the reputation of being very effective and efficient signal analysis tool. Wavelet analysis is capable of retrieving features of data including trends, breakdown points, discontinuities and self similarities.

## 2. DISCRETE WAVELET TRANSFORMATION

In order to detect the minor faults[2] on the transformer winding, DWT is proposed due to it's time and frequency localization property. The DWT is one of the three forms of WT .It moves a time domain discretized signal into it's corresponding wavelet domain. This is done through a process called "sub-band decomposition" performed using digital filter banks.

For a given electrical signal f(n) the spectral bands decomposition[3]is carried out using successive decomposition of signal via pair of High pass and Low pass filter as illustrated in Fig.1.

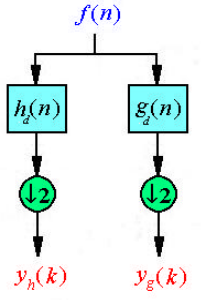


*Fig.1–Sub-band decomposition scheme of a signal*

Basically, the DWT evaluation has two stages. The first consists on the wavelet coefficients determination. These coefficients represent the given signal in the wavelet domain. From these coefficients, the second stage is achieved with the calculation of both the approximated and the detailed version of the original signal, in different levels of resolutions, in the time domain. At the end of the first level of the signal decomposition, the resulting vectors $y_h(k)$ and $y_g(k)$will be, respectively, the level 1 wavelet coefficients of detail and approximation coefficient.

In a similar fashion the calculation of the approximated[4] ($c_A2(n)$) and the detailed ($c_D2(n)$) version associated to the level 2 is based on the level 1 wavelet coefficient of approximation ($c_A1(n)$). The process goes on, always adopting the "n-1" wavelet coefficient of approximation to calculate the "n" approximated and detailed wavelet coefficients. Once all the wavelet coefficients are known, the discrete wavelet transform in the time domain can be determined.

Figure 2 shows the spectral decomposition of a secondary side output for power transformer and the decomposed detail and approximated coefficients. The spectral bands provide the information of disturbances or variable frequency content for the given signal based on which the level of distortion in secondary current can be evaluated.

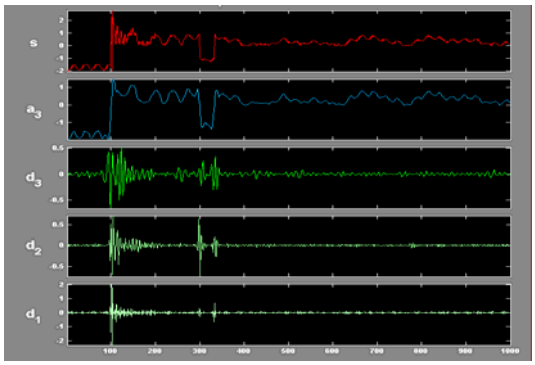


*Fig.2 - wavelet decomposition waveforms*

This resolutional decomposition provides the variations, which are in spectral domain and are not available in spatial domain. These variations can result in more accurate estimation as compared to spatial estimation. To perform this decomposition in real time application filter bank architectures are realized using filter chips (or) DSP processors [5]. Both the approaches provide resolutional information but are large area covering, high power consuming and slower in response due to delay in data transfer . The delay in response delay may result in improper operation of electrical control device resulting in lowering of life-cycle for costly and reliable electrical equipments.

To reduce the above difficulties associated with the traditional DSP Processors or filter chips, it is proposed filled programmable Gate Arrays (FPGA). Technology which offers the potential of designing high performance system at low cost.

## 3. DIGITAL MODELING OF DWT

For the realization of the stated DWT architecture, the filter bank architecture is developed using VHDL coding. The design architecture is as shown in figure 3. The discretized[6] current pulse is passed as input to this system in 16 bit floating represented in excess-7 notation. The samples are buffered into the input FIFO of 16 x 16 location and are passed to the filter bank via buffer logic. The inputs are off-centered by two and are passed as a block of 4 samples per cycle. These samples are buffered into

the buffer logic and are passed to the filter bank on request generation. A pair of High pass and a Low pass filter bank is realized for each level of decomposition.

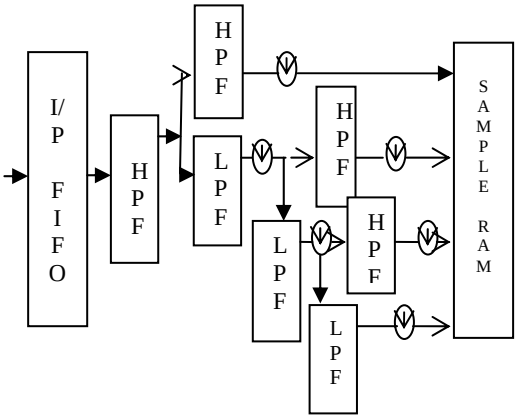


*Fig. 3. Digital architecture realized for Wavelet Transformation*

Each wavelet coefficient is decomposed by a factor of 2 before passing it to the sample RAM. The sample RAM is developed with 12 x 16 location for holding the wavelet coefficient after every high pass filter output.

The filter logics are realized using MAC[7] (multiply and accumulate) operation where a recursive addition, shifting and multiplication operation is performed to evaluate the output coefficients. The recursive operation logic is as shown below.

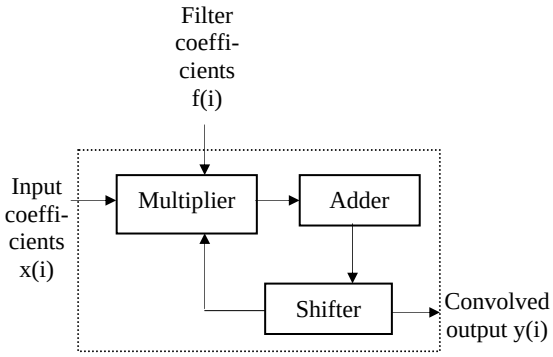


*Fig 4 : Realization of recursive MAC operation*

Before passing the data to filter bank the fifo logic realized stores the data in asynchronous mode of operation, operating on the control signals generated by the controller unit. On a read signal the off-centered data is passed to the buffer logic. The fifo logic is realized as shown below.

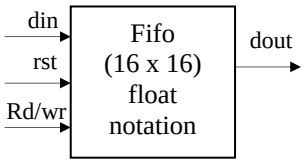


*Fig 5: Realization of 16 x 16 fifo logic for coefficient interface*

The obtained detail coefficients are down sampled by a factor of two to reduce the number of computation inturn resulting in faster operation. To realize the decimator operation comparator logic with a feedback memory element is designed as shown below.

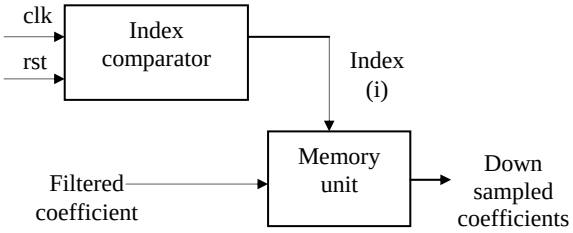


Fig 6: architecture for decimation by 2 logic

## 4. VHDL MODELING TO REALIZE DWT

The proposed system is realized using VHDL language for it's functional definition. The HDL modeling[8] is carried out in top-down approach with user defined package support for floating point operation and structural modeling for recursive implementation of the filter bank logic. For the realization a package is defined with user defined record data type as

```
type  real_single is
     record
      sign : std_logic;
      exp: std_logic_vector(3 downto 0);
     mantissa: std_logic_vector(10 downto 0);
  end record;
```

The floating notation is implemented using 16 bit IEEE-754 standards as presented below.

| Sign. (1) | Exp. (4) | Mantissa (11) |
|---|---|---|

The floating-point addition, multiplication and shifting operation are implemented as procedures in the user defined package and are repeatedly called in the implementation for recursive operation. The procedures[9] are defined as;

```
procedure shifftl (arg1: std_logic_vector;arg2: integer;arg3 :out std_logic_vector);
procedure shifftr (a:in std_logic_vector; b:in integer;result: out std_logic_vector);
procedure addfp (op1,op2: in real_single;op3: out real_single) ;
procedure fpmult (op1,op2: in real_single;op3: out real_single) ;
```

for performing the convolution operation, filter coefficients are defined as constant in this package and are called by name in filter implementation.

```
constant
lpcof0: real_single:=('1',"0100","00001001000");
constant
lpcof1: real_single:=('0',"0100","11001010111");
constant
lpcof2:real_single:=('0',"0110","10101100010");
constant
lpcof3:real_single:= '0',"0101","11101110100");
constant hpcof0:
real_single:= ('1',"0101","11101110100");
constant
hpcof1:real_single:=('0',"0110","10101100010");
constant
hpcof2:real_single:=('1',"0100","11001010111");
constant
hpcof3:real_single:=('1',"0100","00001001000");
```

using the above definitions the filters are designed for high pass and low pass operation. The recursive implementation is defined as;

```
for k in 1 downto 0 loop
        old(k):=shift(k);
    fpmult(old(k)(0),hpf(k+1),pro(k)(0));
        proper(j,k):=pro(k)(0);
    addfp(acc(k)(0),pro(k)(0),acc(k)(0));
        acer(j,k):=acc(k)(0);
    shift(k+1):=shift(k);
 end loop;
```

for the evaluation of the implemented design the test vectors are passed through the test bench generated from Matlab tool. The continuous output of secondary side transformer obtained after impulse test are discretized using matlab tool where each coefficient is converted to 16-bit floating notation and passed to the test bench for HDL interface. The coefficients obtained from the filter bank after convolution is then compared with the results obtained from the matlab decomposition for accuracy evaluation.

```
library ieee;
use work.math_pack1.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_1164.all;
entity topmodule_wb is
end topmodule_wb;
architecture TB_ARCHITECTURE of
topmodule_wb is

component topmodule
        port (
                clk : in std_logic;
                rst : in std_logic;
                start : in std_logic;
                read1 : in std_logic);
        end component;

signal STIM_clk : std_logic;
        signal TMP_clk : std_logic;
        signal STIM_rst : std_logic;
        signal STIM_start : std_logic;
        signal STIM_read1 : std_logic;
        signal WPL  : WAVES_PORT_LIST;
        signal TAG  : WAVES_TAG;
        signal ERR_STATUS: STD_LOGIC:='L';
  begin
       CLOCK_GEN_FOR_clk: process
        begin
        if END_SIM = FALSE then
                TMP_clk <= '0';
                wait for 50 ns;
                 else
                wait;
                end if;
                if END_SIM = FALSE then
                        TMP_clk <= '1';
                        wait for 50 ns;
                else
                wait;
                end if;
        end process;
ASSIGN_STIM_clk: STIM_clk <= TMP_clk;
ASSIGN_STIM_rst:
STIM_rst
<= WPL.SIGNALS(TEST_PINS'pos(rst)+1);
ASSIGN_STIM_start:
STIM_start
<= WPL.SIGNALS(TEST_PINS'pos(start)+1);
UUT: topmodule
        port map(
                 => ,
                clk => STIM_clk,
                rst => STIM_rst,
                start => STIM_start,
                 => ,
```

```
                => ,
            read1 => STIM_read1,
                => );
end TB_ARCHITECTURE;
end TESTBENCH_FOR_topmodule;
```

## 5. RESULT

The sampled input data and the comparison of subsequent wavelet coefficients from MATLAB Program, HDL code is as shown below :

*Input Data :*

| Output from impulse test as input | Digital binary data |
|---|---|
| 0.21751 | ‘0’,”1010”,”11000000001” |
| 0.0158 | ‘0’,”0110”,”00001100001” |
| 0.0.365 | ‘0’,”0101”,”01100101000” |
| 0.0325 | ‘0’,”0100”,”01001110001” |
| 0.01245 | ‘0’,”0110”,”01001100000” |

*Detail Coefficients at level 1* :

| Matlab coeff. | HDL output (Binary) | Decimal equivalent |
|---|---|---|
| 0.15192 | ‘0’,”1001”,”00111100110” | 0.15171 |
| 0.003154 | ‘0’,”0100”,”00010111101” | 0.00312 |
| 0.1245 | ‘0’,”1001”,”01101010110” | 0.1243 |
| 0.22545 | ‘0’,”0111”,”00010100010” | 0.2233 |
| 0.003214 | ‘0’,”1000”,”00011010001” | 0.003113 |

*Detail Coefficients at level 2 :*

| Matlab coeff. | HDL output (Binary) | Decimal equivalent |
|---|---|---|
| 0.211 | ‘0’,”1010”,”01010111000” | 0.211 |
| 0.00124 | ‘0’,”0011”,”00001001001” | 0.00123 |
| 1.0024 | ‘0’,”0100”,”00010111000” | 1.0024 |
| -0.036 | ‘1’,”0110”,”00101100000” | -0.035 |
| -0.02145 | ‘1’,”0100”,”00011111001” | -0.02142 |

*Detail Coefficients at level 3 :*

| Matlab coeff. | HDL output (Binary) | Decimal equivalent |
|---|---|---|
| 0.2245 | ‘0’,”1000”,”00011000110” | 0.225 |
| -0.661 | ‘1’,”0011”,”01000011101” | -0.66 |
| -0.002458 | ‘1’,”0111”,”00010001000” | -0.00232 |
| 0.124 | ‘0’,”0101”,”00011000001” | 0.124 |
| 0.0325 | ‘0’,”0100”,”01000001000” | 0.0323 |

*Approximate Coefficients :*

| Matlab coeff. | HDL output (Binary) | Decimal equivalent |
|---|---|---|
| 0.2254 | ‘0’,”0100”,”01000010001” | 0.2243 |
| -0.0884 | ‘1’,”1001”,”00000111000” | -0.0874 |
| -0.02154 | ‘1’,”0101”,”01000100000” | -0.02122 |
| 0.2245 | ‘0’,”0011”,”10000010001” | 0.2235 |
| 0.45457 | ‘0’,”0110”,”00010011100” | 0.45435 |

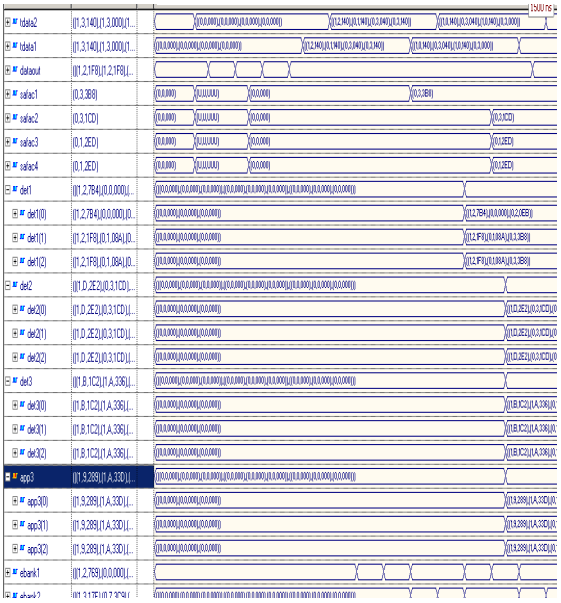

*Fig. 7: Functional simulation result for designed DWT on Aldec’s simulator showing the detail and approximate coefficient obtained after filtration*

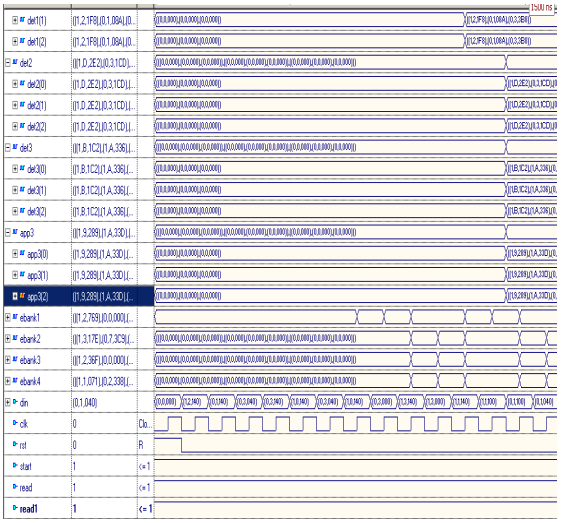

*Fig 8: functional simulation result for designed DWT on Aldec’s simulator showing input’s for the design*

The functional results obtained after the convolution operation carried out for the implemented filter design is shown in figure 7. The coefficients are generated from the discretized samples passed from the test-bench interface where the stimulus are taken from the MATLAB

generated binary coefficients of the secondary side power transformer. The coefficients are compared with resolution coefficients obtained from the MATLAB results and are almost found equal with 0.01 variations, resulting in high accuracy in computation.

About 15 cycles of system clock for performing the operation. This time is comparatively 85-90 % less as compared to the time taken for performing filtration operation in MATLAB simulation. The test vectors are passed through test bench for simulation as illustrated in figure 8.

## 6. FPGA REALIZATION

The designed system is targeted onto xilinx xc2vpx70-7-ff1704 FPGA device belonging to virtex2p family with a speed grade of –7. The implementation of deigned DWT processor is illustrated in figure 9. The[(10)] logical routing can be observed from the obtained Place and route result form the FPGA Editor option in xilinx synthesizer. It is observed that about 40% area for the targeted FPGA is covered for the implementation of DWT processor. Figure 10 shows the logical utilization in each configurable logical blocks (CLB) in the implemented FPGA. The CLB's are connected in cascade manner to obtain the functionality for the designed processor.

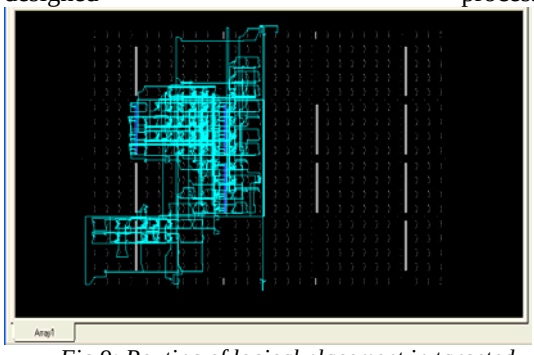

*Fig 9: Routing of logical placement in targeted (xc2vpx70-7-ff1704) FPGA*

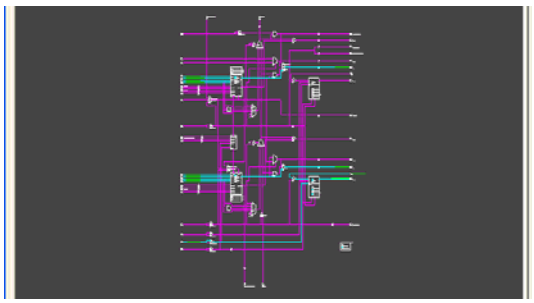

*Fig 10. Logical utilization of CLB in targeted FPGA*

The synthesis result for the designed DWT processor is presented

Macro Statistics

| | |
|---|---|
| # Registers | : 49 |
| # Multiplexers | : 25 |
| # Tristates | : 74 |
| # Adders/Subtractors | : 618 |
| # Multipliers | : 29 |
| # Comparators | : 128 |
| Design Statistics # IOs | : 26 |
| Cell Usage : # BELS | : 181 |
| Minimum period | : 5.220ns |
| (Maximum Frequency | : 191.571MHz) |

From the result it is observed that a logical count of 181 Basic element logic (BEL) are required for the realization of DWT processor. The real time Maximum operating frequency obtained is 191.571 MHz. This operation frequency is considerably higher than the current sample frequency and make it more suitable for real time current analysis.

The power analyzer of xilinx tool is used for the evaluation of power consumption and thermal summary for the designed DWT processor for real time operation. from the report generated the power consumed is about 204 mW under operating condition with working temperature of 25C, which are very suitable under real time implementation.

Part : 2vp100ff1696-6
Data version : ADVANCED,v1.0,05-28-03

| Power summary : | I(mA) | P(mW) |
|---|---|---|
| Total estimated power consumption : | | 204 |
| Vccint 1.50V : | 100 | 150 |
| Vccaux 2.50V : | 20 | 50 |
| Vcco25 2.50V : | 2 | 4 |

Thermal summary:

| | |
|---|---|
| Estimated junction temperature : | 25C |
| Ambient temp : | 25C |
| Case temp : | 25C |

The Register transfer logic (RTL) implementation for the designed processor is shown in figure 11.

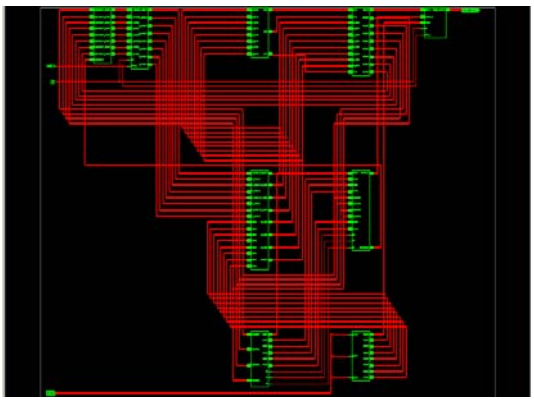

*Fig 11. RTL implemented for the designed DWT processor.*

## 7. CONCLUSION

FPGA implementation for DWT processor for the analysis of power transformer faults is realized. The implementation of DWT processor on FPGA results in high speed operation of automated power quality analyzer by replacing the existing filter bank architecture (or) DSP based architecture resulting in more reliable operations for power quality analysis. The implementation results obtained from xilinx synthesizer shows a very low resource utilization with high speed real time operating frequency and low power consumption ,with ambient temperature condition which are most suitable for real time installation in power quality analysis. The developed FPGA design could be merged with advanced learning standards for the total automation of fast and reliable power transformer protections in electrical power system. This facility leads to the concept of reconfigurability, which is advantageous and not high resource consuming under remote applications.

## REFERENCES


[1]. R. MLEWSKY, *"Five Years of Monitoring the Impulse Test of Power Transformer with Digital Recorders and Transfer Function Method"*, pp.1-6,CIGRE 1992 .session 12-20.

[2]. W. Wang , Y.M.Li , Y.Qui *" Application of Wavelet Analysis to Detection of Transformer Winding Deformation, 10th ISH'* 97, pp 131-134 Montreal Qubec Canada 1997.

[3]. Peter Hoffman AND Surya Santoso, *" Power Quality Assessment via Wavelet Transform Analysis"*, IEEE Transaction on Power Delivery Vol,.11,No 2 April 1996.

[4]. Daubechies, I. (1990) *" The wavelet transform, time/frequency location and signal analysis. IEEE Transactions on Information Theory*", 36, 961-1005.

[5]. S.Masud *"VLSI system for discrete wavelet transforms"*, PhD Thesis, Dept. of electrical engineering, The Queen's University of Belfast, 1999.

[6]. Rioul, O. and M. Vetterli (1991) *" Wavelets and signal processing. IEEE Signal Processing Magazine"*, 14-38.

[7]. Vetterli, M. and J. Kovacevic (1995) *" Wavelets and Subband Coding"*. Prentice Hall, Englewood Cliffs, NJ, U.S.A.

[8]. HDL Designer Series User Manual, Software Version 2003.1,9 April 2003, Mentor Graphics Corporation 1996-2003.

[9]. Modelsim 5.6 SE Performance Guidelines, Model Technology February 2002, User's Manual, Version 5.6e, Mentor Graphics Corporation 1996-2002.

[10]. *"XC4000E and XC4000X Series Field Programmable Gate Arrays"*, Xilinx 1999.