A Novel Way to Design and Implement Statistical Operations based on FPGA

Sarmad F. Ismael
Computer Science Department
Cihan University
Iraq – Duhok

Basil Shukr Mahmood, PhD
Computer Engineering Department
University of Mosul
Mosul – Iraq

ABSTRACT
The architecture design for statistical operations to compute the Mean, Variance, Standard Deviation, RMS (Root Mean Square), Covariance, and MSE (Mean Square Error) values has been implemented on hardware concerning Xilinx Spartan 3E XC3S500E FPGA and worked properly up to maximum frequency of 73.252 MHz . The practical outcomes have been compared with the theoretical values calculated by Matlab with maximum error of 1.425%. New methods of design were concerned for the architecture of each function to reduce the number of slices.

General Terms
Computer Architecture, computational technology, Statistic theory

Keywords
FPGA, VHDL, Statistical Operations, Accumulators, fixed point.

1. INTRODUCTION
The most commonly statistical used operations were mean, variance, standard deviation, RMS, Covariance, and MSE. The statistical operations were used in a large number of applications such as digital signal processing [1], image proceeding [2] and many other applications.

In this paper, a novel way is presented to implement the architecture design to do the statistical operations. There are many researches deal with statistical operations, but most of them don't refer to their architectures using FPGA.

In 2008, three methods of accumulation have been examined to find the mean and variance to adapt them to be used in FPGA applications by David B. Thomas and Wayne Luk[3]. In 2009, a floating-point accumulator for FPGA-based high performance computing applications is proposed, evaluated and implemented in FPGA-based by Song Sun and Joseph Zambreno [4]. In 2010, a floating-point accumulation was Performed on a modern FPGA in Single and Double Precision by Tarek Ould Bachir and Jean-Pierre David [5].

The paper is organized as follows: In section (II), the background and theory of the architectural design is presented. In section (III), the hardware Implementation details are shown. In section (IV), the experimental result is given. Accordingly, the section (V) concludes this paper.

2. BACKGROUND AND THEORY
The statistical operations [6] presented in this paper are as follow:

A .mean value (or average):

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad (1)$$

B. Variance value:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 \qquad (2)$$

C. Standard Deviation value:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2} \qquad (3)$$

D. Root Mean Square value:

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} X_i^{\,2}} \qquad (4)$$

E. Mean Square Error value:

$$MSE(x,y) = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2 \qquad (5)$$

F. Covariance value:

$$c = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y}) \qquad (6)$$

To find the variance, standard deviation and covariance values firstly find the mean and then use it in the above equations. These operations need two iterative steps, therefore their expressions have been analyzed as follows:

$$s^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} X_i^{\,2} - \frac{1}{n}\left(\sum_{i=1}^{n} X_i\right)^2\right] \qquad (7) \qquad (7)$$

$$MSE = \frac{1}{n}\left[\sum_{i=1}^{n} X_i^{\,2} - 2\sum_{i=1}^{n} X_i Y_i + \sum_{i=1}^{n} Y_i^{\,2}\right] \qquad (8)$$

$$c = \frac{1}{n-1}\left[\sum_{i=1}^{n} X_i Y_i - \frac{1}{n}\sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i\right] \qquad (9)$$

3. HARDWARE IMPLEMENTATION DETAILS
The general design of the proposed architecture to compute the statistical operations is described in Fig. 1. Using the above manipulated equations, all operations are share with the same accumulators, therefore the components to compute the accumulator operations has been implemented as shown in Fig.1.

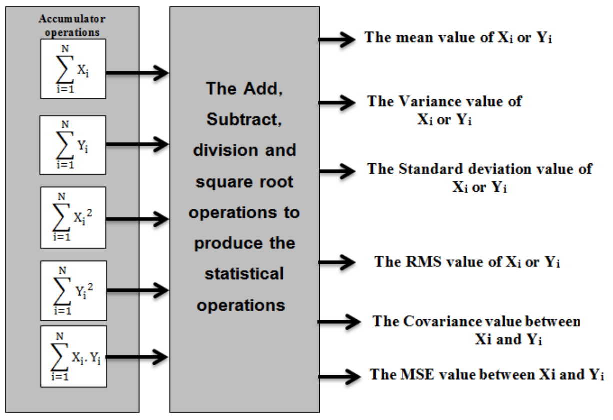

Accumulator operations
$\sum_{i=1}^{N} X_i$
$\sum_{i=1}^{N} Y_i$
$\sum_{i=1}^{N} X_i^2$
$\sum_{i=1}^{N} Y_i^2$
$\sum_{i=1}^{N} X_i.Y_i$
The Add, Subtract, division and square root operations to produce the statistical operations
The mean value of Xi or Yi
The Variance value of Xi or Yi
The Standard deviation value of Xi or Yi
The RMS value of Xi or Yi
The Covariance value between Xi and Yi
The MSE value between Xi and Yi

Fig 1: General Component of Statistical operations Architecture

After computing the values of all accumulators for the all expressions, the arithmetic operations such as Add, Subtract, Shift, and Square Root are needed to produce the final results. The arithmetic operations used in the proposed architecture are:

3.1 Accumulators*:*

The Accumulators used in the design represent the main operations in the proposed architecture are of two types: the general accumulator (GAC) and the multiplier accumulator (MAC) as shown in Figure 2 [4][5]:

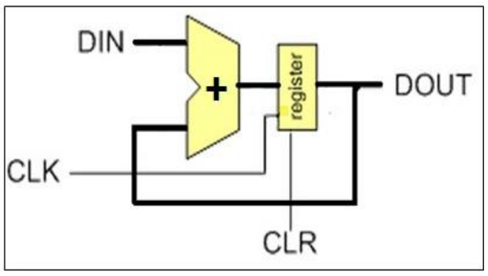


Fig.2 (a) GAC

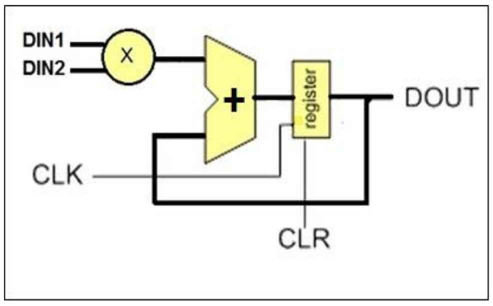


Fig.2 (b) MAC

Fig.2. Types of accumulator used in the design.

Clear (CLR) signal in the accumulator is used to clear the output in order to receive new input data.

3.2 Addition and Subtraction:

As these operations are ready in VHDL, their codes were copied to the proposed design.

3.3 Multiplier:

Characterized by low size and very high speed, the 18X18 Embedded Multipliers in Spartan 3E XC3S500E FPGA have been used.

3.4 Division:

The division operation cannot be implemented directly by using VHDL, therefore many algorithms are introduced for the implementation. The best one of them is the non-restoring algorithm [7] which is suitable for the large number of accumulator bits. Since the large number of bits in the accumulator usually needs large number of slices in design, the number of input data fixed to 16 inputs and shift to right is used to represent division operation instead of using division algorithm. Shift to right 4 bits to represent division by N and the conjugate multiplication method have been suggested to simplify the denominator N-1 as show in Eq. 10.

$$\frac{1}{N-1}*\frac{N+1}{N+1}=\frac{N+1}{N^2-1}*\frac{N^2+1}{N^2+1}=\frac{N^3+N^2+N+1}{N^4-1}\cong\frac{1}{N}+\frac{1}{N^2}+\frac{1}{N^3}+\frac{1}{N^4} \qquad (10)$$

The accumulator result is shifted to right four times (4 bits , 8 bits , 12 bits , 16 bits), and then add them to produce the result.

3.5 Square Root:

The modified non-restoring algorithm has been used to implement square root operation which is depending on subtract operation and append 01 and need one clock cycle to produce the result [8].

Fig. 3 shows the design of the proposed architecture.

The proposed architecture consists of two main parts, the first part is to calculate the five accumulator operations and to input data X and Y which work in parallel, therefore they need just 16 clocks to complete all values and then they can be used many times to produce all statistical operations. The second one contains many 2x1 multiplexers (MUX) to select which input X or Y is used as well as many arithmetic operations that have been discussed before.

The reset bit has been used to synchronously reset all accumulator operations when new next input data acquired. Two types of MUX have been used, the first one is 2X1 MUX to select whether the input data is X or Y, depending on the first bit of the selector signal. The second multiplexer 8X1 MUX is used to select which statistical operation has to appear on the output depending on the last three bits of the selector. Table 1 shows which statistical operation is selected according to the input selector signal. The remaining operations are add, sub, shift right and shift left have been used to accomplish the remaining statistical operations.

The fixed point package [9] is used to represent variables in the VHDL code. Table 2 shows the length and the format of each variable used in the design. The output of the statistical operations assumed to have 24 bit lengths, therefore they are padded to be 24 bit length.

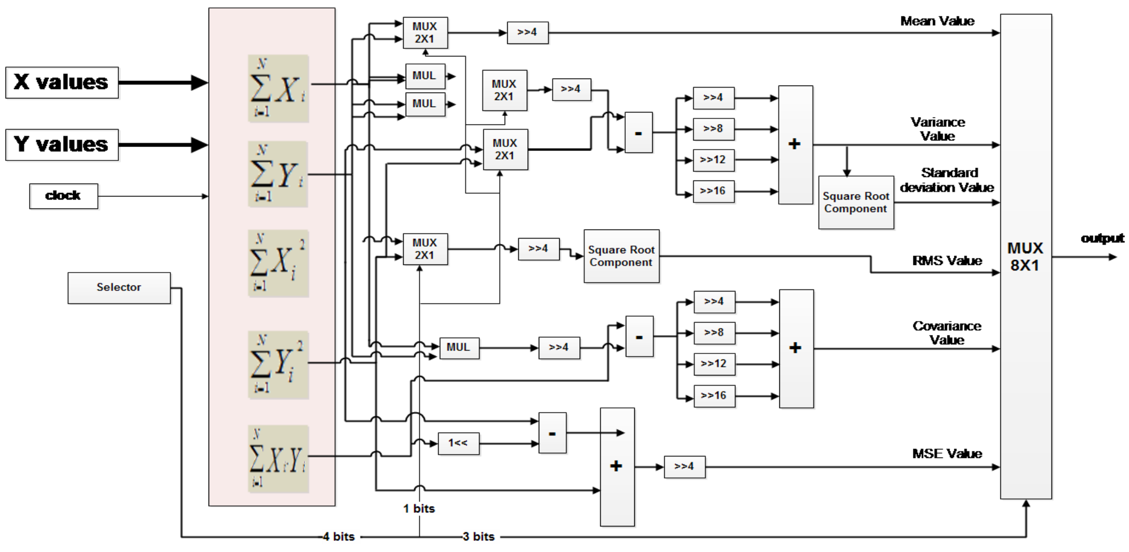


Fig.3. The proposed architecture of statistical operations.

Table 1. Statistical operations selector according to input selector signal

Selector (3)	Selector (2)	Selector (1)	Selector (0)	statistical operations
0	0	0	0	Mean (X)
0	0	0	1	Mean(Y)
0	0	1	0	Variance(X)
0	0	1	1	Variance (Y)
0	1	0	0	Standard Deviation(X)
0	1	0	1	Standard Deviation(Y)
0	1	1	0	RMS(X)
0	1	1	1	RMS(Y)
1	0	0	-	Covariance(X,Y)
1	0	1	-	MSE(X,Y)

Table 2. The numbers of bits representation for each element

operations	NO. of real part + sign bit	NO. of fraction part
mean	5	8
variance	12	12
Standard Deviation	6	10
RMS value	5	11
Covariance	12	12
MSE value	13	11

4. EXPERIMENTAL RESULTS

The input data for X and Y first saved in the distributed memory array of Spartan 3E as 16 elements each 9 bits (sign bit , 4 bits for real part and 4 bits for fraction part).

Signed fixed numbers have been used to represent X and Y values. table 3 shows examples of X and Y values with their hex_decimal representations.

Table 3. X and Y values used as inputs

X Values	Y Values	HEX for X	HEX for Y
3.1	2.3	032	025
-1.6	1.9	1E6	01E
1.12	-1.1	012	1EE
-0.91	-0.81	1F1	1F3
-10.111	0.78	15E	00C
0.4	0.515	006	008
2.2	0.629	023	00A
-5.6	-3.23	1A6	1CC
-12.7	11.6	135	0BA
3.8	-15.313	03D	10B
4.66	-2.9	04B	1D2
-1.222	0.1	1EC	002
10.61	-0.09	0AA	1FF
0.99	0.13	010	002
-1.31	8.9	1EB	08E
9.87	4.2	09E	043

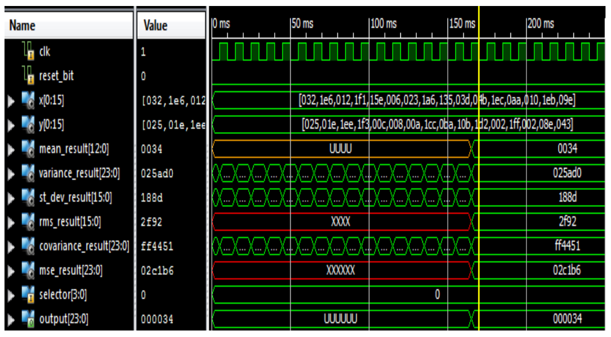
After feeding X and Y values inputs, the values of all the statistical operations will appear on the output after 16 clock cycles, the simulation results of all the statistical operations coming out from the ISE 14.7 simulator are shown in Figure 4.

The theoretical values calculated by Matlab for the same input values have been tested and compared with the experimental practical values taken, and the error values between them are shown in table 4.

Error values calculated based on Eq. 11 from the table 4 shows that the maximum error produced is not greater than 1.425 %

$$Error = \frac{Theoretical\ \ value - Practical\ \ value}{Theoretical\ \ value} * 100 \qquad (11)$$

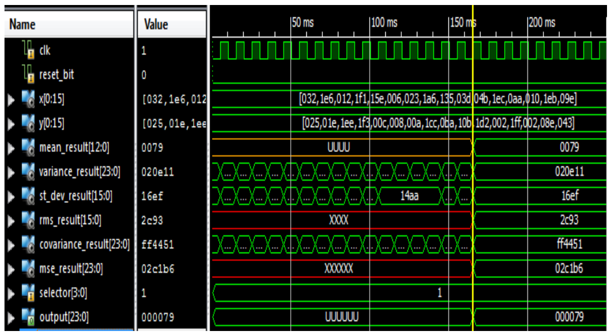
The complete architecture design was implemented on Xilinx Spartan 3E XC3S500E FPGA by using ISE14.7 simulator. After synthesizing the design by using ISE14.7 simulator, it is found that the maximum operating frequency becomes 73.252 MHz and the summary of resources utilization from the FPGA chip is shown in table 5.



Name	Value
clk	1
reset_bit	0
x[0:15]	[032,1e6,012
y[0:15]	[025,01e,1ee
mean_result[12:0]	0034
variance_result[23:0]	025ad0
st_dev_result[15:0]	188d
rms_result[15:0]	2f92
covariance_result[23:0]	ff4451
mse_result[23:0]	02c1b6
selector[3:0]	0
output[23:0]	000034

0 ms
50 ms
100 ms
150 ms
200 ms
[032,1e6,012,1f1,15e,006,023,1a6,135,03d,04b,1ec,0aa,010,1eb,09e]
[025,01e,1ee,1f3,00c,008,00a,1cc,0ba,10b,1d2,002,1ff,002,08e,043]
UUUU
0034
025ad0
188d
XXXX
2f92
ff4451
XXXXXX
02c1b6
0
UUUUUU
000034

Fig.4 (a)



Name	Value
clk	1
reset_bit	0
x[0:15]	[032,1e6,012
y[0:15]	[025,01e,1ee
mean_result[12:0]	0079
variance_result[23:0]	020e11
st_dev_result[15:0]	16ef
rms_result[15:0]	2c93
covariance_result[23:0]	ff4451
mse_result[23:0]	02c1b6
selector[3:0]	1
output[23:0]	000079

50 ms
100 ms
150 ms
200 ms
[032,1e6,012,1f1,15e,006,023,1a6,135,03d,04b,1ec,0aa,010,1eb,09e]
[025,01e,1ee,1f3,00c,008,00a,1cc,0ba,10b,1d2,002,1ff,002,08e,043]
UUUU
0079
020e11
14aa
16ef
XXXX
2c93
ff4451
XXXXXX
02c1b6
1
UUUUUU
000079

Fig.4 (b)

Fig.4. The simulation results of all statistical operations: (a) For the input X – values, (b) For the input Y – values

Table 4. Theoretical and practical values with errors for each statistical operation

Operations	Theoretical result		Practical result		ERROR %
	Decimal	HEX	Decimal	HEX	
Mean(a)	0.206062	35	0.203125	034	1.425%
Mean(b)	0.475687	7A	0.472656	079	0.637%
Variance(a)	37.586	25960	37.6758	025aD0	0.238%
Variance (b)	32.8794	20E12	32.8792	020E11	0.0006%
Standard Deviation(a)	6.13074	1886	6.1377	188d	0.113%
Standard Deviation(b)	5.73405	16F0	5.7334	16ef	0.011%
RMS(a)	5.93964	2F84	5.94629	2f92	0.111%
RMS(b)	5.57232	2C94	5.57178	2c93	0.0096%
Covariance (a,b)	-11.7573	FFA1F1	-11.7302	FF4451	0.23%
MSE(a,b)	88.1788	2C16E	88.2139	2c1b6	0.0398%

Table 5. The summary of resources utilization

Logical Utilization	Used	Available	Utilization
Number of Slices	1090	4656	23 %
Number of Slice Flip Flops	220	9312	2 %
Number of 4 input LUTs	1988	9312	21 %
Number of bonded IOBs	30	232	12 %
Number of MUL18X18	6	20	30 %
Number of GCLKs	1	24	4 %

5. CONCLUSIONS AND FUTURE WORK

The proposed architecture produces six statistical operations for two inputs X and Y with better performance by using a small area and high frequency of Spartan 3E chip and few clock cycles. The used area of Spartan 3E is 23% and the maximum operating frequency becomes 73.252MHz. This enhancement in performance is due to the usage of novel way to deal with the statistical equations and the use of shift operations instead of using division operation which needs more time, therefore 16 clock cycles are needed to produce all statistical operations, this means that results of all statistical operations are calculated by less than 220 nsec. The proposed architecture is useful in real time applications as well.

In future works, the proposed architecture can be used to design a statistical processor in order to solve statistical problems in many science fields that require these operations.

6. REFERENCES

[1] S. K Smith, United States of America: Newnes, 2003, ch.2 Digital Signal Processing.

[2] Sallee, Philip Andrew, 2004 Statistical methods for image and signal processing, PhD diss., UNIVERSITY OF CALIFORNIA DAVIS.

[3] D.B. Thomas and W. Lu, FPT 2008, Estimation of sample mean and variance for Monte-Carlo simulations. In ICECE Technology, International Conference on, pp. 89-96. IEEE.

[4] S. Sun and J. Zambreno, 2009 A Floating-point Accumulator for FPGA-based High Performance Computing Applications. IEEE International Conference in Field-Programmable Technology, Pages: 493 – 499.

[5] T. O. Bachir and J. P. David. 2010 performing floating-point accumulation on a modern FPGA in single and double precision. Field-Programmable Custom Computing Machines (FCCM), 18th IEEE Annual International Symposium on. IEEE.

[6] R. D. Mason, D. A. Lind and W. G. 1998 Marchal. Statistics: an introduction. Duxbury Pr.

[7] B.Jovanovic and M. Jevtic November 2010 FPGA Implementation of Throughput Increasing Techniques of The Binary Dividers, international scientific conference, Page: 397-401 19 – 20.

[8] T. Sutikno, A. Z. Jidin, A.Jidin and N. R. Idris, March 2012 Simplified VHDL Coding of Modified Non-Restoring Square Root Calculator , International Journal of Reconfigurable and Embedded Systems (IJRES), Vol. 1, No. 1,pages 37-42.

[9] D.Bishop, 1076-2008 Fixed point package user's guide. Packages and bodies for the IEEE.