CENTRAL INSTITUTE OF TECHNOLOGY, KOKRAJHAR

(Centrally Funded Institute under MHRD, Govt. of India)

KOKRAJHAR::783370:: BODOLAND

Estd. :: 2006

A

Project Report

On

SINE AND COSINE FUNCTION GENERATOR USING VHDL

Submitted by,

DHARMESWAR BORO

ROLL NO: Gau-c-10/L-322.

PINOSH KR HAJOARY

ROLL NO: Gau-c-10/L-336.

MUNGSHAR BORO

ROLL NO: - Gau-c-10-267.

Table of contents:

CANDIDATE DECLARATION:

We certify that this project titled "**A Project Report On SINE AND COSINE FUNCTION GENERATOR Using VHDL**", a perquisite towards partial fulfillment for the award of Bachelor of Technology (B.TECH) in engineering, is an accurate record of our work carried under the guidance of Mr.Kaushik Chandra Deva Sarma, project supervisor and member of Electronics And Communication Engineering department of CIT.

THE ABOVE IS BEST OF OUR KNOWLEDGE AND IT IS TRUE

DHARMESWAR BORO (Gau-c-10/L-322)

Dharmeswar Boro

PINOSH KUMAR HAJOARY(Gau-c-10/L-336)

MUNGSHAR BORO(Gau-c-10-267)

Mungshar Boro

CERTIFICATE FROM THE GUIDE:

I do hereby approve and forward the project titled "**A Project Report On SINE AND COSINE FUNCTION GENERATOR Using VHDL** " carried out by Dharmeswar Boro, Pinosh Kr. Hajoary and Mungshar Boro and certifies that the above declaration made by the candidates is correct to the best of my knowledge.

....

(Mr. Kaushik Chandra Deva Sarma)

Asst. Prof. ECE department

ACKNOWLEDGEMENT:

First of all we would like to thank those who have supported us to bring out this text within a short period of time.

Especially we are very much thankful to Mr. Kaushik Chandra Deva Sarma and Mr. Antaryami Panigrah (Asst. Prof. Electronics and Communication Engineering) for providing this golden opportunity to work out on this project and make this successful.

Last but not the least we would like to express our sincere heartful gratitude to our hostel mates for helping us from the beginning of this project and make it a grant success.

Place: CIT KOKRAJHAR, KOKRAJHAR

Date : 21/11/13

DHARMESWAR BORO (Gau-c-10/L-322)

Dharmeswar Boro

PINOSH KR HAJOARY(Gau-c-10/L-336)

MUNGSHAR BORO(Gau-c-10-267)

Abstract:

This project is concerned with the design of a 32-bit digital sine and coisne function generator for modern application specific computer using the combined scheme algorithm. A trigonometric-wave generator circuit is provided to generate data representative of a wave of trigonometric functions by using two ROMs (which stores one half of the sine function period of only a selected phase region between 0 and π) and some adders. Moreover, in order to regenerate a cosine wave, sine designation data are used to cooperate with the high-order bits of the input phase data. Thus, the circuit is capable of generating sine-wave data or cosine-wave data correctly with high precision.And by constructing a divider circuit we can generate other trigonometric functions with the help of these sine and cosine wave data values and by applying CORDIC (Co-ordinate Rotation digital Computer) algorithm. CORDIC is an iterative algorithm for the calculation of the trigonometric functions using only adder and shift operations.
This project presents architecture for designing a flexible and scalable digital trigonometric function wave generator.

An FPGA-Based architecture is presented and the design has been implemented on a Xilinx ISE9.1i using VHDL. Synthesis and simulation results are shown and discussed.

Chapter 1: Itroduction:

As the project is concerned with the design of a 32-bit sine and cosine function wave generator for modern application specific computer using the combined scheme algorithm. Let us first introduce what is a combined scheme. A combined scheme is a superposition of some specific schemes viz. sine function calculation scheme, lookup table based generator scheme and oscillation scheme with single and multiple frequencies.

Now, let us LEARN what these schemes are -

In sine function calculation scheme we use the different approximation algorithms like Taylor scheme, CORDIC algorithm, interpolation algorithm, etc. to determine the sine and cosine function values.

The idea in lookup table-based generator is to build a table of M samples of the sine function, which form a single sine wave period ($S = \mathrm{Sin}(2\pi i/M)$).

Oscillation scheme with single frequency is an algorithm, which generates the sine waves using some fundamental properties of transcendental functions. It is usually implemented as the solving of some difference equation as
$y(i) = 2\cos(b)y(i\text{-}1) - y(i\text{-}2), i=0,1,$...............

Such a scheme generates the sine wave by the initial conditions

$y(-1)= -\sin(b); y(-2)= -\sin(2b);$

or the cosine wave by the conditions

$y(-1)= \cos(b); y(-2)= \cos(2b);$

with the frequency $f = bf_s/(2\pi)$ Hz.

Oscillation scheme with multiple frequency is an algorithm based on well-known trigonometric formulas: Sin(A+B) = SinACosB + CosASinB; and Cos(A+B) = CosACosB – SinASinB;

where SinA,CosA are samples of the generated waves and B is the angle to which the neighboring samples differ (it represents the given frequency).

Each and all the above mentioned schemes can generate sine and cosine function wave values individually. But the values of the sine and cosine function generated by each of them are not precise and accurate. Thus, we super impose all the above mentioned schemes to get more precise and accurate trigonometric function values.

Chapter 1.1: General Discussion on the applied Algorithm:

(1) Sine Function Calculation:

Sine function calculation is the usual way to generate sine waves in PC and other program controlled computers. Here we use the different approximation algorithms like Taylor scheme, CORDIC algorithm, interpolation algorithm, etc. For example, sine and cosine functions at the interval |x|<1 can be estimated as

$$\sin(\pi x/2) = 57063x - 64323x^3 + 07271x^5.$$
$$\cos(\pi x/2) = 9994 - 22279x^2 + 22399x^4. \quad \ldots\ldots\ldots\ldots\ldots(1)$$

with the error, which is less than 06%.

The disadvantage of this method consists in the large complexity of calculations (in the example above – up to 6 multiplications and 2 additions for the sine function). Besides, here the functions are defined for the angles less than $\pi/2$, and additional calculations are needed for deriving the functions in another ranges. The advantage is a wide range of the generated frequencies.

(2) Look Up Table based Generator:

The idea in lookup table-based generator is to build a table of M samples of the sine function, which form a single sine wave period. That means that at the address i the value $S = \sin(2\pi i/M)$ is stored. The wave generation means reading the samples, addressing them by the incremented address counter. The increment k=1,2,...,M/2 of such a counter is proportional to the resulting sine wave frequency $f = kf_S/M$, where f_S is the sampling frequency. The precisions of both the outputted sine wave and its frequency installing depend on the table volume M and the data width of the coefficients S.

(3) Oscillation Scheme:

Oscillation scheme with single frequency is an algorithm, which generates the sine waves using some fundamental properties of transcendental functions. It is usually implemented as the solving of some difference equation. For example, the following difference equation

$$y(i) = 2\cos(b)y(i-1) - y(i-2); \text{ where } i=0,1,\ldots\ldots\ldots \quad \ldots\ldots\ldots\ldots(2)$$

models the second order recursive digital filter at the border of amplification and excitation modes. Such a scheme generates the sine wave by the initial conditions

$$y(-1)= -\sin(b); y(-2)= -\sin(2b); \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

or the cosine wave by the conditions

$$y(-1)= \cos(b); y(-2)= \cos(2b);$$

with the frequency $f = bf_S/(2\pi)$ Hz. Theoretically equation (2) represents the stable sine wave generator, i.e. it operates without damping or saturation of oscillations, if the multiplier of $y(i-2)$ (if any) is equal precisely to a 1. But the sine and cosine coefficients must be truncated by the machine representation in such a way, that the sine of a zero angle to be equal to a zero. i.e. $y(0) = 2\cos(b)(-\sin(b)) - (-\sin(2b)) = 0.$ ……………………………(4)

The disadvantages of this generator are: -small scope of the frequency regulating, which is limited by the data bit width and relation (4); - wave magnitude is different for the different frequencies, and some displacement is present (see the table below); - for a set of frequencies the set of coefficients must be calculated and/or stored; - very high ($>0.13f_S$) and very low ($<0.001f_S$) frequencies could not be generated without extreme errors.

Chapter 1.1: General Discussion on the applied Algorithm:

Oscillation scheme with multiple frequencies is based on the well-known trigonometric formulas: $\sin(x+y) = \sin x \cos y + \cos x \sin y$;

$\cos(x+y) = \cos x \cos y - \sin x \sin y$. .... (5)

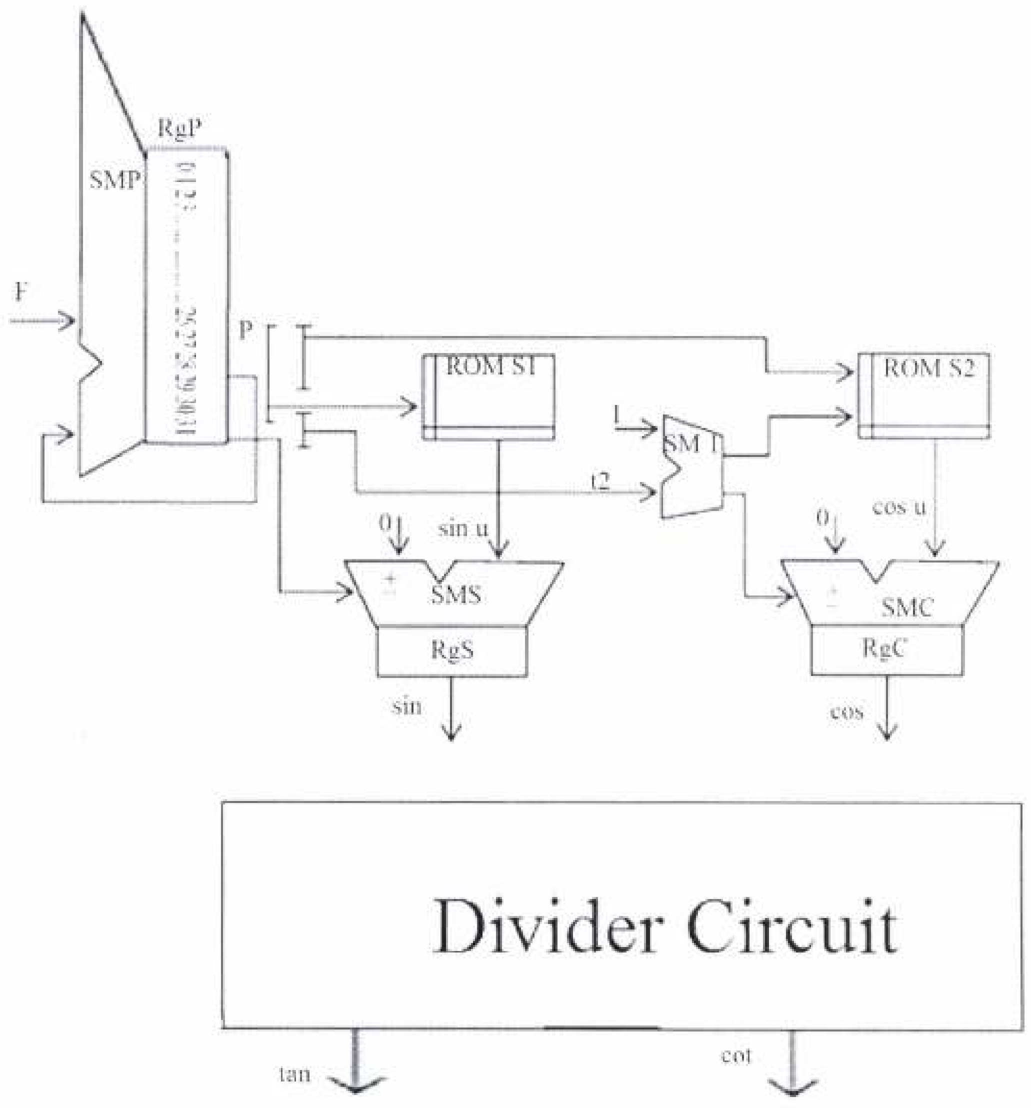
Here $\sin x$, $\cos x$ are samples of the generated waves, and y is the angle to which the neighboring samples are differ, i.e. it represents the given frequency. The disadvantage of this sheme consists in its unstability due to the unprecise representation and calculation of the sine and cosine samples. That means that $\sin^2 x + \cos^2 x \neq 1$ $\sin^2 y + \cos^2 y \neq 1$ due to the truncation errors in equation(2) . This feature can be minimized by addition of some nonlinearities to this scheme which will decrease the increased signal magnitude.

(4) Combined Scheme:

In the combined schemes the superposition of the mentioned above schemes is used. For example, consider the generator which frequency must be tuned precisely. Then such generator can be built as two generators, one of them generates sine and cosine waves with the high frequency and another one does them with the low frequency. The resulting signal is derived by the mixing the signals of both of them using the above equations (5).

Chapter 2.: Circuit Model:

Chapter 2.1: Circuit Diagram:



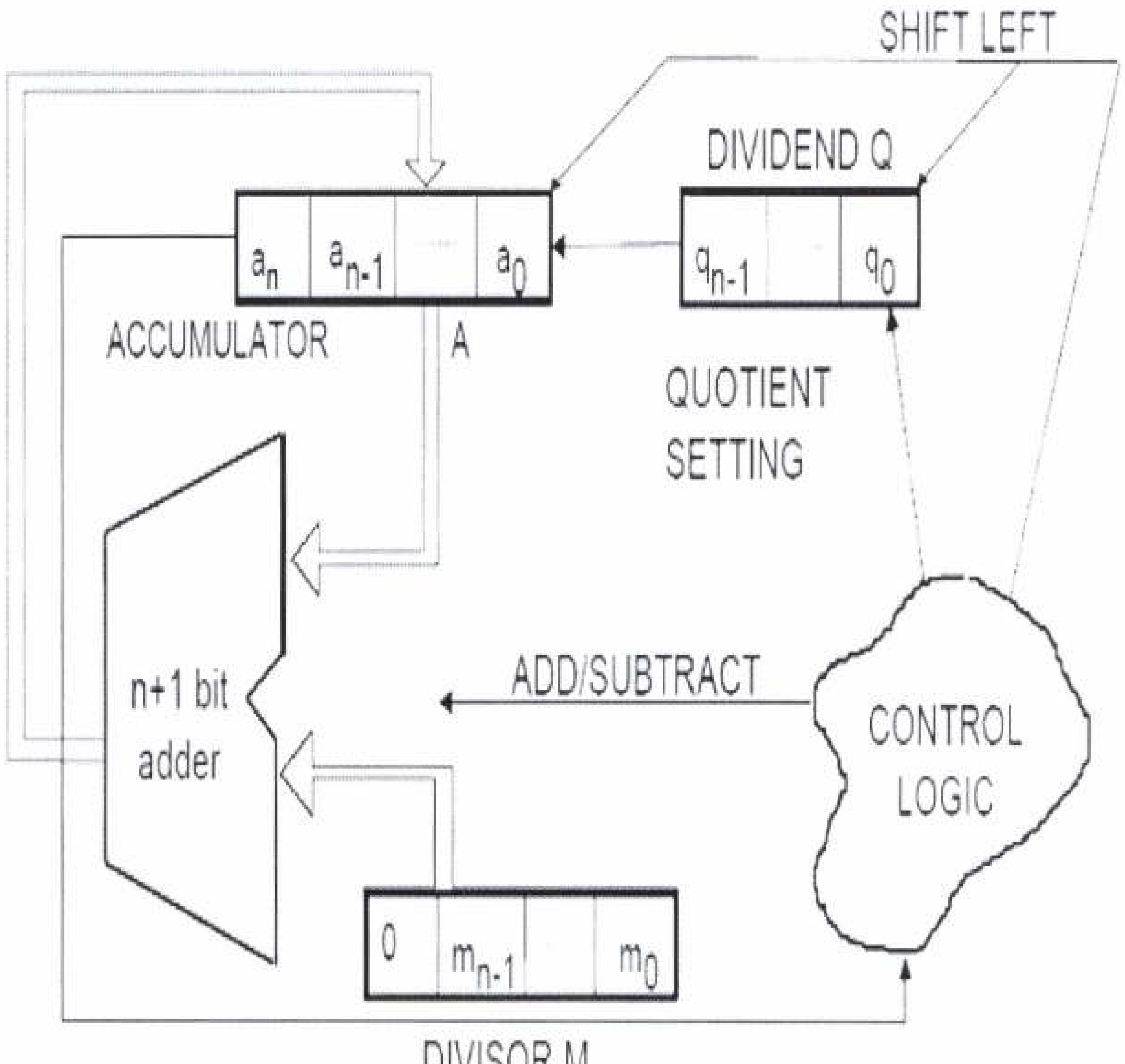
Chapter 2.2 Description of the circuit:

In the above circuit we took one 32-bit input data (F), four adders (SMP,SMS, SMC and SM1), three registers (RgP, RgS and RgC) to store data and two ROMs (ROM S1 and ROM S2). The adder SMP with the register RgP implements the phase accumulator with the increment *F*. ROM ROMS1, ROMS2 store one half of the sine function period. ROMS1 is addressed by 5 most significant bits of RgP except highest one. Therefore, the whole sine wave period consists of 32 samples. The adder SMS inverts the sine code to generate the negative waves of the resulting sine signal. It is implemented when the 15th bit of RgP is a 1. In another situation this adder throughputs the data without exchanges.

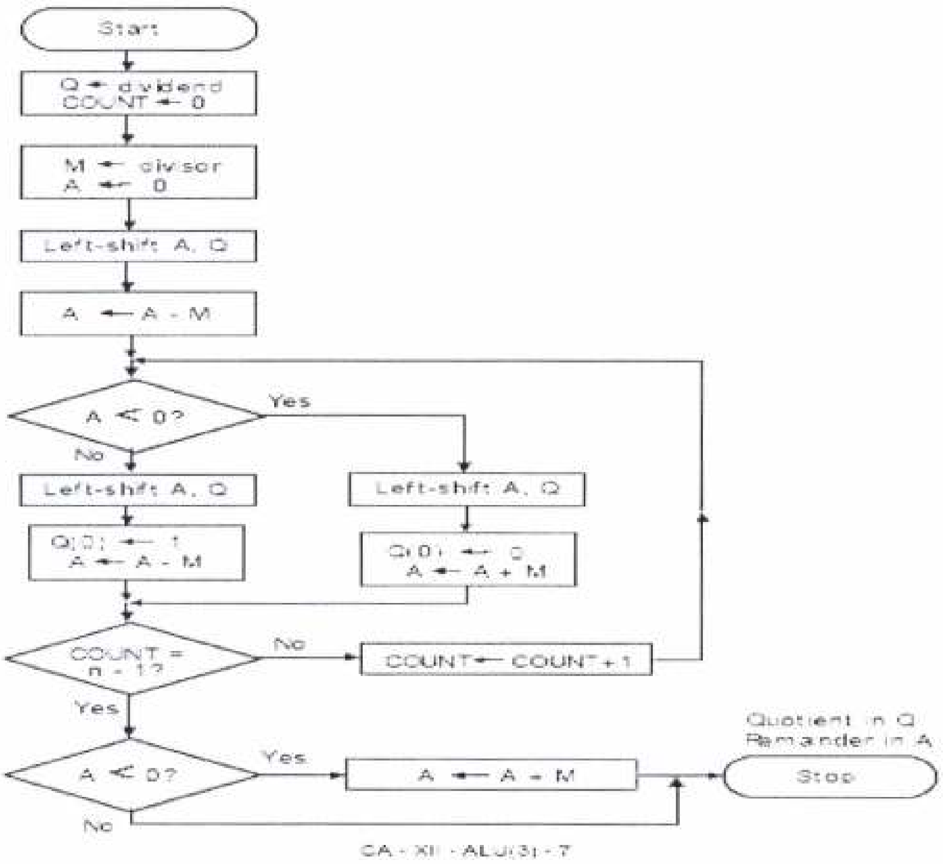
The adder SM1 adds a 1 to 2 MSBs of the code RgP, and therefore, it shifts the phase value to 90°. By this method the address is derived which provides the cosine function fetching from the sine table. The adder SMC and ROMS2 generate the cosine function. Sine and cosine samples are buffered in the registers RgS and RgC respectively.

The sine and cosine values stored in registers RgS and RgC are then divided in a divider circuit to obtain the tangent and cotangent values.

Chapter 2.3: Divider Algorithm:



Chapter 2.4:Flow chart of the Divider:



Algorithm for division:

- Set Count to 0 and put 0 in A register.
- Start loop for n times
- Shift A & Q left one binary position
- Subtract M from a, placing the answer back in A
- if the sign of A < 0, set Q0 to 0 and add M back to A (restore A);
- otherwise, set q0 to 1
- Check for count, when count = n-1 then stop the loop.

Chapter 3: Advantages and Disadvantages:

Advantages:

1> Cost effective.
2> Low power consumption.
3> Low size.
4> High speed operation.
5> Simple and easy to implement.
6> It is a computer integrated.
7> Relatively less truncation error.

Disadvantages:

1> Unstable due to imprecise representation and calculation of sine and cosine samples (that means $\sin^2 x + \cos^2 x$ /= 1)
2> It is has complex operation.
3> Prone to damage due to high power.

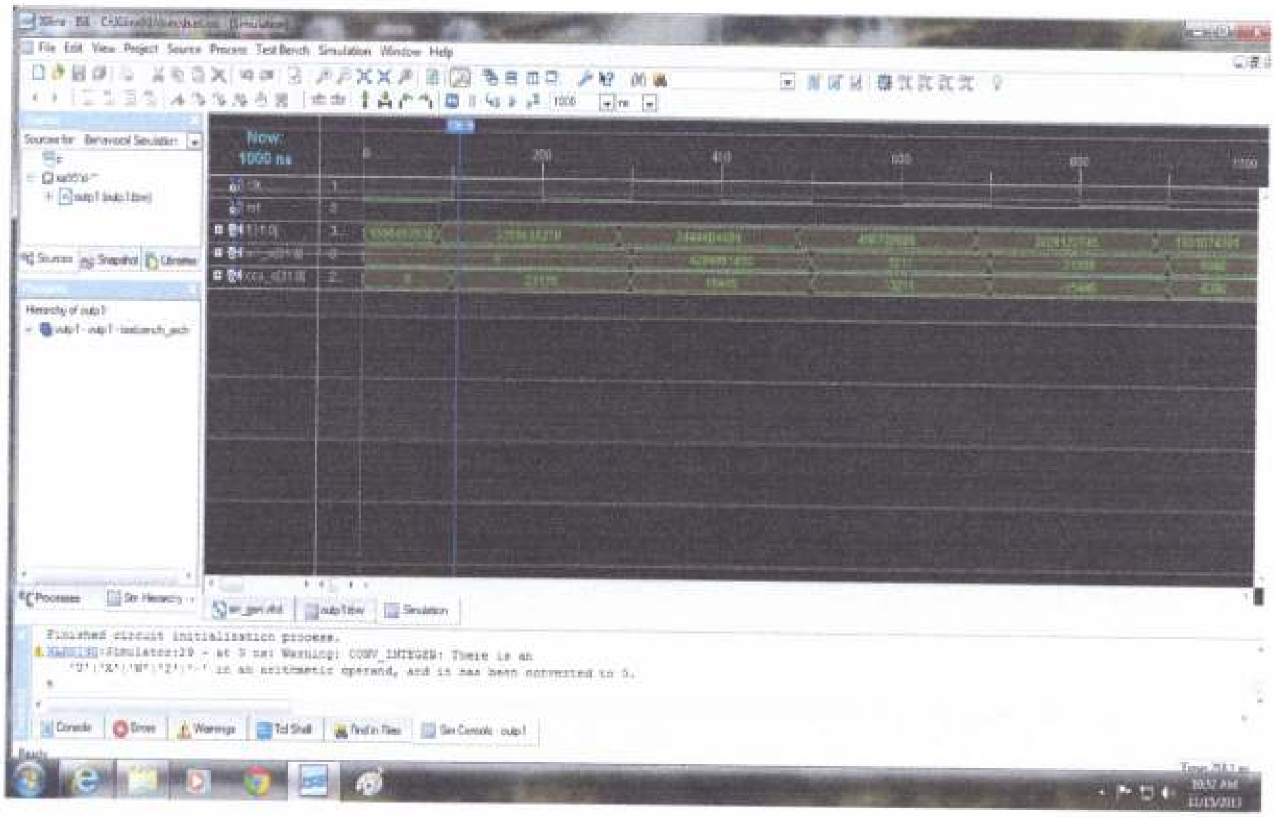
Chapter 4 : Application:

Application:

Trigonometric functions have a wide range of uses including computing unknown lengths and angles in triangles for instance, in navigation, engineering, and physics for frequency conversion, discrete Fourier transform, in modems, software defined radios, radars, mobile phones, radio receivers, etc.

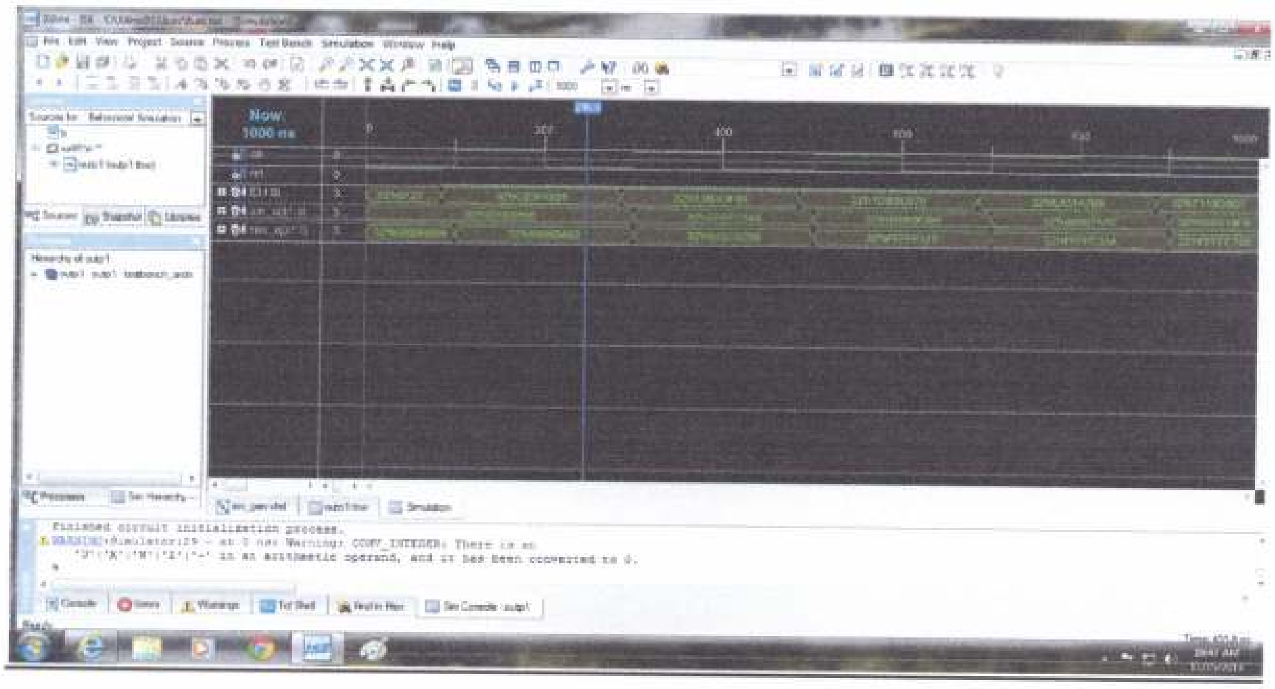
The sine and cosine functions are also commonly used to model periodic function phenomena such as sound and light waves, the position and velocity of harmonic oscillators, sunlight intensity and day length, and average temperature variations through the year.

Chapter 5: **Result:**



```
Finished circuit initialization process.
WARNING:Simulator:29 - at 0 ns: Warning: CONV_INTEGER: There is an
   'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, and it has been converted to 0.
```

Fig 1: Simulation result in Decimal form.



```
Finished circuit initialization process.
WARNING:Simulator:29 - at 0 ns: Warning: CONV_INTEGER: There is an
   'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, and it has been converted to 0.
```

Fig 2: Simulation result in Hexa-decimal form.

Chapter 5: **Result and Schematic view:**

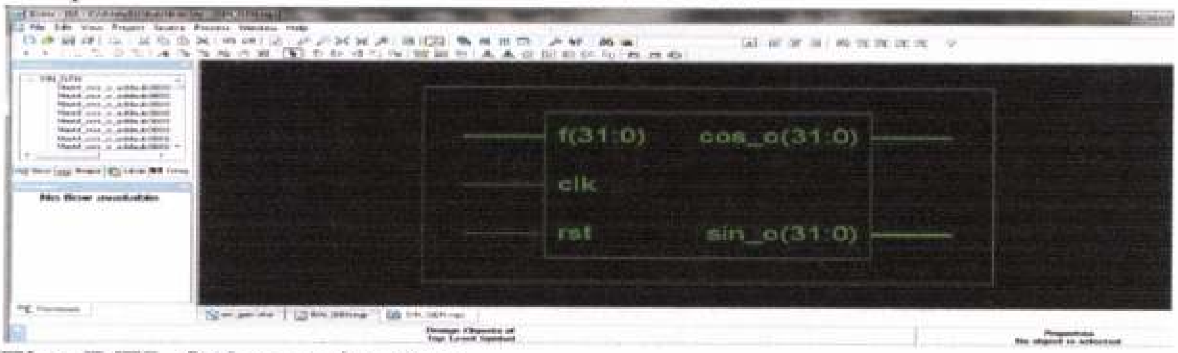


Fig: RTL Schematic view.

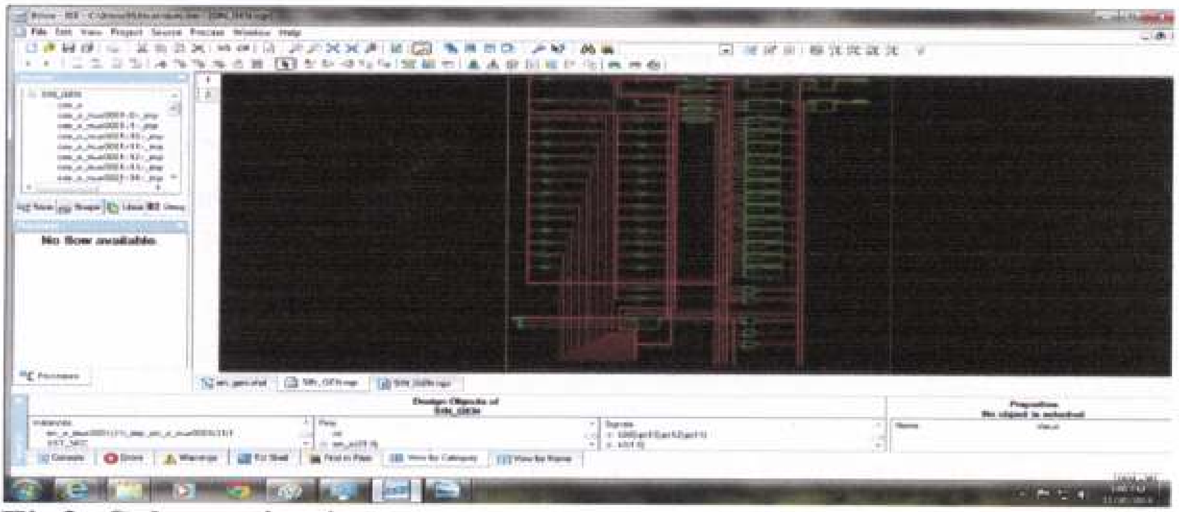


Fig2: Schematic view.

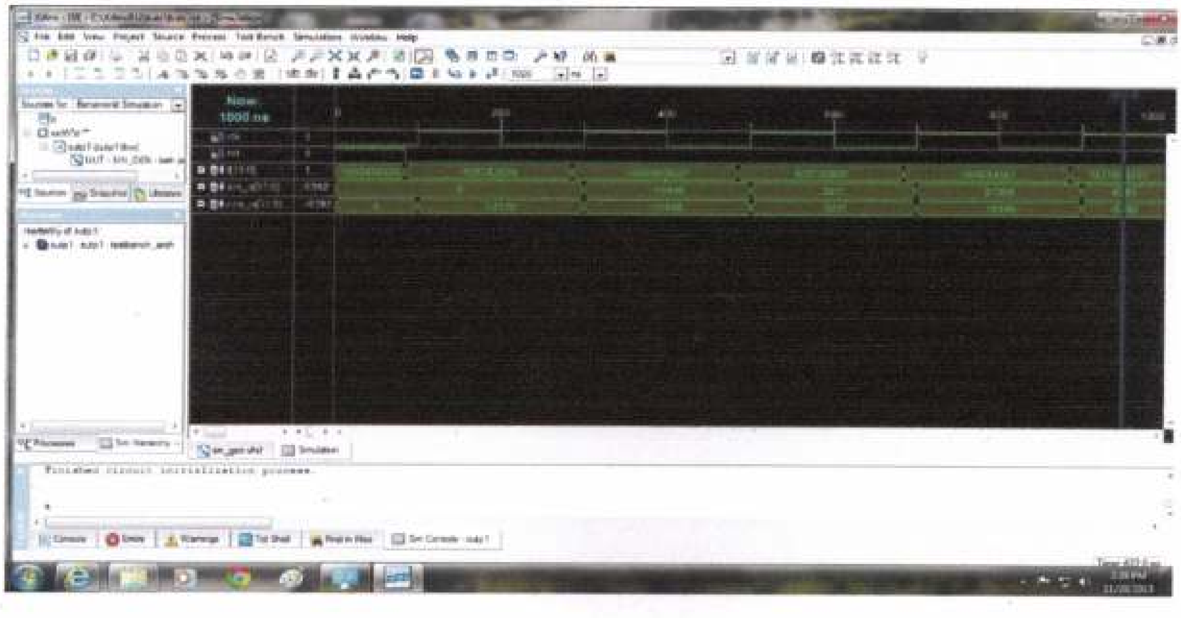


Fig 3: Simulation result in Decimal form.

Chapter 6: Conclusion:

In this project we discussed various schemes of generating trigonometric function wave such as sine function calculation, lookup table-based generators, oscillation scheme with single frequency and multiple frequencies and combined scheme, the disadvantages associated with them and proposed a new model to overcome those disadvantages by simply super imposing all the schemes and take the output from the mixture following the condition given by the relation of the trigonometric formula given by equation(5)

(i.e. $\sin(A+B) = \sin A \cos B + \cos A \sin B$; and $\cos(A+B) = \cos A \cos B - \sin A \sin B$).

Thus, a high precision and cost effective trigonometric function wave generator can be constructed by using combined scheme algorithm.

Chapter 7: Future course of work:

We will design a divider circuit to find other trigonometric function values (tangent, cotangent, secant and cosecant) using CORDIC algorithm. We would further upgrade the generator from 32-bit and to 64- bit to get more precision and accuracy in the result.

Chapter 8: Appendix:

8.1.Timing Report :

Design Name	SIN_GEN
Device, Speed (Speed File Version)	XCR3256XL, -7 (6.0)
Date Created	Tue Nov 12 15:09:32 2013
Created By	Timing Report Generator: version J.30
Copyright	Copyright (c) 1995-2007 Xilinx, Inc. All rights reserved.

Summary

Notes and Warnings
Note: This design contains no timing constraints.
Note: A default set of constraints using a delay of 0.000ns will be used for analysis.

Performance Summary	
Min. Clock Period	142.800 ns.
Max. Clock Frequency (fSYSTEM)	7.003 MHz.
Limited by Cycle Time for clk	
Clock to Setup (tCYC)	142.800 ns.
Setup to Clock at the Pad (tSU)	141.100 ns.
Clock Pad to Output Pad Delay (tCO)	4.500 ns.

Timing Constraints

Constraint Name	Requirement (ns)	Delay (ns)	Paths	Paths Failing
TS1000	0.0	0.0	0	0
AUTO_TS_F2F	0.0	142.8	814	814
AUTO_TS_P2P	0.0	4.5	64	64
AUTO_TS_P2F	0.0	142.1	529	529
AUTO_TS_F2P	0.0	3.5	64	64

Constraint: TS1000

Description: PERIOD:PERIOD_clk:0.000 nS

Path	Requirement (ns)	Delay (ns)	Slack (ns)

Constraint: AUTO_TS_F2F

Description: MAXDELAY:FROM:FFS(*):TO:FFS(*):0.000 nS

Path	Requirement (ns)	Delay (ns)	Slack (ns)
p_0_0.Q to p_0_30.D	0.000	142.800	-142.800
p_0_0.Q to p_0_31.D	0.000	142.800	-142.800
p_0_1.Q to p_0_30.D	0.000	142.800	-142.800

Constraint: AUTO_TS_P2P

Description: MAXDELAY:FROM:PADS(*):TO:PADS(*):0.000 nS

Path	Requirement (ns)	Delay (ns)	Slack (ns)
clk to cos_o<0>	0.000	4.500	-4.500
clk to cos_o<10>	0.000	4.500	-4.500
clk to cos_o<11>	0.000	4.500	-4.500

Constraint: AUTO_TS_P2F

Description: MAXDELAY:FROM:PADS(*):TO:FFS(*):0.000 nS

Path	Requirement (ns)	Delay (ns)	Slack (ns)
f<0> to p_0_30.D	0.000	142.100	-142.100
f<0> to p_0_31.D	0.000	142.100	-142.100
f<1> to p_0_30.D	0.000	142.100	-142.100

Constraint: AUTO_TS_F2P

Description: MAXDELAY:FROM:FFS(*):TO:PADS(*):0.000 nS

Path	Requirement (ns)	Delay (ns)	Slack (ns)
cos_o<0>.Q to cos_o<0>	0.000	3.500	-3.500
cos_o<10>.Q to cos_o<10>	0.000	3.500	-3.500
cos_o<11>.Q to cos_o<11>	0.000	3.500	-3.500

Number of constraints not met: 4

Data Sheet Report:Maximum External Clock Speeds

Clock	fEXT (MHz)	Reason
Clk	7.003	Limited by Cycle Time for clk

Chapter 9: Synthesis Report:

TABLE OF CONTENTS:

1>Synthesis Options Summary :

Source Parameters	
Input File Name	: "SIN_GEN.prj"
Input Format	: mixed
Ignore Synthesis Constraint File	: NO
Target Parameters	
Output File Name	: "SIN_GEN"
Output Format	: NGC
Target Device	: CoolRunner XPLA3 CPLDs
Source Options	
Top Module Name	: SIN_GEN
Automatic FSM Extraction	: YES
FSM Encoding Algorithm	: Auto
Safe Implementation	: No
CASE Implementation Style	: Full-Parallel
Mux Extraction	: YES
Resource Sharing	: YES
Target Options	
Add IO Buffers	: YES
MACRO Preserve	: YES
XOR Preserve	: YES
Equivalent register Removal	: YES
General Options	
Optimization Goal	: Speed
Optimization Effort	: 1
Library Search Order	: SIN_GEN.lso
Keep Hierarchy	: YES

RTL Output	: Yes
Hierarchy Separator	: /
Bus Delimiter	: <>
Case Specifier	: maintain
Verilog 2001	: YES
Other Options	
Clock Enable	: YES
wysiwyg	: NO

2>HDL Compilation:

Compiling vhdl file "C:/Xilinx91i/btec1/sin_gen.vhd" in Library work.

Entity <sin_gen> compiled.

WARNING:HDLParsers:1406 - "C:/Xilinx91i/btec1/sin_gen.vhd" Line 46. No sensitivity list and no wait in the process

Entity <sin_gen> (Architecture <beh>) compiled.

3> **Design Hierarchy Analysis** :

Analyzing hierarchy for entity <SIN_GEN> in library <work> (architecture <beh>).

4> HDL Analysis:

Analyzing Entity <SIN_GEN> in library <work> (Architecture <beh>).

WARNING:Xst:790 - "C:/Xilinx91i/btec1/sin_gen.vhd" line 64: Index value(s) does not match array range, simulation mismatch.

WARNING:Xst:790 - "C:/Xilinx91i/btec1/sin_gen.vhd" line 66: Index value(s) does not match array range, simulation mismatch.

Entity <SIN_GEN> analyzed. Unit <SIN_GEN> generated.

5> HDL Synthesis :

Performing bidirectional port resolution...

Synthesizing Unit <SIN_GEN>.

Related source file is "C:/Xilinx91i/btec1/sin_gen.vhd".

WARNING:Xst:646 - Signal <t> is assigned but never used.

```
WARNING:Xst - Property "use_dsp48" is not applicable for this technology.
WARNING:Xst - Property "use_dsp48" is not applicable for this technology.
    Found 16x16-bit ROM for signal <$mux0000> created at line 64.
    Found 16x16-bit ROM for signal <$mux0001> created at line 66.
    Found 32-bit register for signal <sin_o>.
    Found 32-bit register for signal <cos_o>.
    Found 32-bit adder for signal <cos_o$addsub0000> created at line 81.
    Found 32-bit up accumulator for signal <p>.
    Found 32-bit adder for signal <sin_o$addsub0000> created at line 76.
    Found 2-bit adder for signal <t2>.
    Summary:
        inferred   2 ROM(s).
        inferred   1 Accumulator(s).
        inferred   3 Adder/Subtractor(s).
Unit <SIN_GEN> synthesized.
WARNING:Xst - Property "use_dsp48" is not applicable for this technology.
```

5.1> **HDL Synthesis Report:**

```
Macro Statistics
# ROMs                          : 2
 16x16-bit ROM                  : 2
# Adders/Subtractors            : 3
 2-bit adder                    : 1
 32-bit adder                   : 2
# Accumulators                  : 1
 32-bit up accumulator          : 1
# Registers                     : 2
 32-bit register                : 2
```

6> Advanced HDL Synthesis:

6.1>Advanced HDL Synthesis Report:

Macro Statistics	
# ROMs	: 2
16x16-bit ROM	: 2
# Adders/Subtractors	: 3
2-bit adder	: 1
32-bit adder	: 2
# Accumulators	: 1
32-bit up accumulator	: 1
# Registers	: 64
Flip-Flops	: 64

7>Low Level Synthesis :

INFO:Xst:2261 - The FF/Latch <cos_o_15> in Unit <SIN_GEN> is equivalent to the following 16 FFs/Latches, which will be removed : <cos_o_16> <cos_o_17> <cos_o_18> <cos_o_19> <cos_o_20> <cos_o_21> <cos_o_22> <cos_o_23> <cos_o_24> <cos_o_25> <cos_o_26> <cos_o_27> <cos_o_28> <cos_o_29> <cos_o_30> <cos_o_31>

INFO:Xst:2261 - The FF/Latch <sin_o_15> in Unit <SIN_GEN> is equivalent to the following 16 FFs/Latches, which will be removed : <sin_o_16> <sin_o_17> <sin_o_18> <sin_o_19> <sin_o_20> <sin_o_21> <sin_o_22> <sin_o_23> <sin_o_24> <sin_o_25> <sin_o_26> <sin_o_27> <sin_o_28> <sin_o_29> <sin_o_30> <sin_o_31>

Optimizing unit <SIN_GEN>

8> Partition Report:

Partition Implementation Status

No Partitions were found in this design.

9> Final Report

Final Results:

RTL Top Level Output File Name : SIN_GEN.ngr

```
Top Level Output File Name      : SIN_GEN
Output Format                   : NGC
Optimization Goal               : Speed
Keep Hierarchy                  : YES
Target Technology               : CoolRunner XPLA3 CPLDs
Macro Preserve                  : YES
XOR Preserve                    : YES
Clock Enable                    : YES
wysiwyg                         : NO
Design Statistics
# IOs                           : 98
Cell Usage :
# BELS                          : 745
#      AND2                     : 261
#      AND3                     : 18
#      AND4                     : 3
#      INV                      : 209
#      OR2                      : 141
#      OR3                      : 16
#      XOR2                     : 97
# FlipFlops/Latches             : 64
#      FDC                      : 64
# IO Buffers                    : 98
#      IBUF                     : 34
#      OBUF                     : 64
CPU : 5.99 / 6.32 s | Elapsed : 6.00 / 6.00 s
Total memory usage is 178268 kilobytes, Number of warnings :    7(   0 filtered)
```

Chapter 10: References:

1. *The Student's Guide to VHDL,2nd Edition* by Peter J. Ashenden ; Morgan Kaufmann Publishers (an imprint of Elsevier).
2. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.Vol. 2, Issue 7, July 2013.*
3. Suhas D Kakde, S S Mane, " *VHDL IMPLEMENTATION OF DIGITAL SINE COSINE GENERATOR USING CORDIC ALGORITHM* ", Inventi Impact: VLSI , Vol. 2013 , Article ID " Inventi : evs/82/13 " , 2013
4. *Sine-wave generator circuit*; US-Patent 5631586 Issued on 1997.
5. S. F. Obennan and M. J. Flynn, "Division algorithms and implementations," IEEE Transactions on Computers, vol. 46, pp. 833- 854, 1997.
6. D. Perry, VHDL, New York, NY: McGraw Hill, Second Edition, 1994.
7. IEEE Standard VHDL Language Reference Manual : ANSI/IEEE Std 1076-1993, New York : IEEE June 1994.
8. R. Andrata, "A survey of CORDIC algorithms for FPGA based computers," *Proc. of the 1998 ACM/SIGDA Sixth Inter. Symp. on Field Programmable Gate Arrays (FPGA '98)*, pp. 191-200, Monterey, CA, Feb. 1998.
9. J. Cao, B. W. Y. Wei, and J. Cheng, "High-performance architectures for elementary function generation," *Proc. of the IEEE Symp. on Computer Arithmetic (ARITH'01),*, Vail, Co, pp. 136-144 , June 2001.
10. D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a line or its caricature," *The Canadian Cartographer*, Vol. 10, No. 2, pp. 112-122, 1973.
11. H. Hassler and N. Takagi, "Function evaluation by table Loo k-up and addition," *Proc. of the 12th IEEE Symp. on Computer Arithmetic (ARITH'95)*, Bath, England, pp. 10-16, July 1995.
12. Y. Iguchi, T. Sasao, and M. Matsuura , "Realization of multiple-output functions by recon¯gurable cascades," *International Conference on Computer Design: VLSI in Computers and Processors (ICCD01)*, Austin, TX, pp. 388-393, Sept. 23-26, 2001.
13. D.U. Lee, Wayne, Luk, J. Villasenor, and P. Y. K. Cheung, "Non-uniform segmentation for hardware function evalu ation," *Proc. Inter. Conf. on Field Programmable Logic and Applications*, pp. 796-807, Lisbon, Portugal, Sept. 2003
14. D.U. Lee, Wayne, Luk, J. Villasenor, and P. Y. K. Cheung, "A hardware Gaussian noise generator for channel code evaluation," *Proc. of the 11th Annual IEEE Symp. on Field-Programmable Custom Computing Machine*