ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

Word serial architecture of CORDIC

Mrs. Supriya Pradeep Kurlekar

Assistant professor SITCOE,Yadrav

Abstract : CORDIC is an acronym for Co-ordinate Rotation Digital Computers and was derived by Volder[1] in late 1950's for the purpose of calculating trigonometric functions. It is widely used in the computing of elementary functions and digital signal processing applications[4], particularly where large amounts of rotation operations are necessary.The original algorithm describes the rotation of 2-D vector which can be applied in applications such as DSP (for Fourier Transforms, Digital Filters)[10] computer graphics and Robotics.
CORDIC processing offers high computational rates making it attractive to applications such as computer graphics where a combination of scaling and rotations are required in real time. CORDIC is also attractive to Robotics[7] since the fundamental operation is co-ordinate transformation[2]. However it could be used for more computationally intensive processes such as motion planning [3]and collision detection.Array imaging [3]typically involves complex signal processing[8] which may require many computationally intensive matrix operations[8].
As intended by Jack E. Volder [1] the CORDIC Algorithm only performs shift and add operations and is therefore easy to implement and resource friendly. However, when implementing the CORDIC algorithm one can choose between various design methodologies and must balanced circuit complexity with respect to performance. It avoids the use of traditional multiplier and accumulator unit [MAC unit] which generally is the bottleneck for the faster systems.This paper attempts to explorer FPGA implementation of CORDIC algorithm using word serial architecture.

Keywords: CORDIC,FPGA,DSP,MAC

I. INTRODUCTION

The digital signal processing landscape has long been dominated by microprocessors with special addressing modes and multiply accumulates instructions while these processors are low cost and offers extreme flexibility. They are often not fast enough for truly demanding DSP tasks. The advent as re-configurable logic computers permits the higher speeds as dedicated hardware solutions at low costs that are competitive with traditional software approach. While hardware efficient solutions often exist, the dominance of software systems has kept those solutions out of the spotlight. Much hardware efficient algorithms exists.
Among these algorithms is a class of shift add algorithms collectively known as CORDIC.
CORDIC is an acronym for Co-ordinate Rotation Digital Computers and was derived by Volder[1] in late 1950's for the purpose of calculating trigonometric functions. It is widely used in the computing of elementary functions and digital signal processing applications, particularly where large amounts of rotation operations are necessary.
The original algorithm describes the rotation of 2-D vector which can be applied in applications such as DSP (for Fourier Transforms, Digital Filters) computer graphics and Robotics.
CORDIC processing offers high computational rates making it attractive to applications such as computer graphics where a combination of scaling and rotations are required in real time. CORDIC is also attractive to Robotics since the fundamental operation is co-ordinate transformation. However it could be used for more computationally intensive processes such as motion planning and collision detection.
Array imaging typically involves complex signal processing which may require many computationally intensive matrix operations.
Increasing the complexity of the imaging model places greater demands on accuracy solution to such complex systems requires better and hence more complex algorithms. Most of these algorithms are based on matrix factorization (decomposition) techniques of which singular value decomposition (SVD) is the most robust method. The SVD factorization requires two sided transformation which involves several trigonometric operations and rotations ideally suited to dedicated VLSI hardware (CORDIC Processing) for real time calculations.
Co-ordinate Rotation Digital Computer is a special purpose computer, in this a unique computing technique is employed which is especially suitable for solving trigonometric relationships involved in plane co-ordinate rotation and conversion from rectangular to polar co-ordinate.
Two basic CORDIC modes are known leading to the computation of different functions, the rotation mode and the vectoring mode.
For both modes algorithm can be realized as an iterative sequence of additions/subtraction and shift operations, which

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

are rotations by a fixed rotation angle (sometimes called micro rotations) but with variable rotation direction. Due to the simplicity of involved operations the CORDIC algorithm is very well suited for VLSI Implemetation.

II WORD SERIAL ARCHITECTURE OF CORDIC

As already mentioned CORDIC algorithm can take two primary architectures, namely Word Parallel or Word Serial. In the first part of this chapter we have explained implementation of Word Parallel CORDIC structure. A Word Serial processor minimizes hardware requirements by utilizing a single CORDIC unit repeatedly. The CORDIC iterative equations yields the block diagram [1]shown in fig.

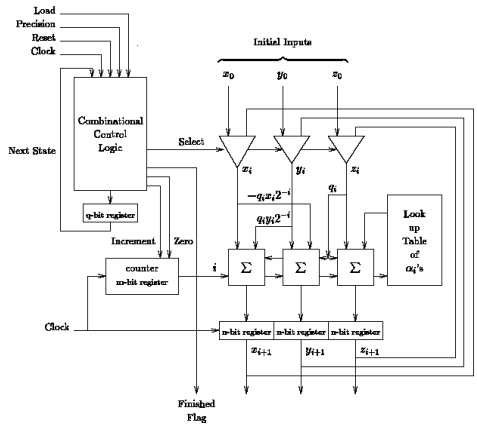


fig 1word serial architecture

II. DESIGN ASPECTS

As shown in word serial architecture we need to implement following blocks-

1.Basic CORDIC block ,containing adder-subtractor, look-up table, sign-control logic for generating decision bit and shifter.

2.Control-logic, for performing iterations ,giving feedback from x,y and z registers, shifting according to iteration ,for doing all this a state machine is designed as control-unit.

The basic CORDIC unit is designed same as in Word Parallel scheme. Controlling hardware is required ,obviously a simple state machine is required to keep track of the current iteration, and to select the degree of shift and ROM addresses for each iteration.

The single stage CORDIC block is implemented same as in unrolled version, same look-up table of arctan values is made. Adder –subtractor logic is developed using add-sub function. The shifter is separately designed in association with each state and each iteration the shifting is carried out. Simple For-Loop logic is written. After each iteration instead of new pipe generation feedback is given to the same block. To perform all these activities Control-Unit is designed using state-machine.

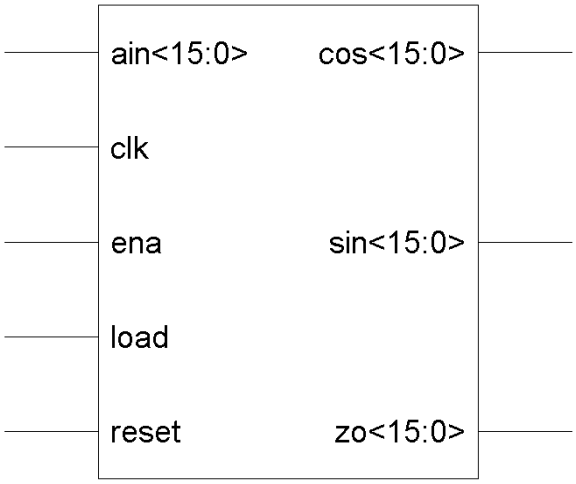
Control Unit

The block required containing the state machine design is the control unit. Control unit is used to keep track of all iterations performed by CORDIC unit and to select degree of shift and ROM address for each iteration.

Control unit has total 16 states from s0 to s15. When ‘reset’ is asserted state machine goes into s0 (reset state) in this state control unit waits for ‘load’ signal when load gets asserted it goes into the next state ‘s1’. In ‘s1’ control unit will assert select signal that goes to the input muxes. Mux will pass the input data to mail CORDIC unit when input sata gets completely loaded into main CORDIC unit and when load signal is released control unit goes into next state ‘s2’. Where it will release select signal, from s2 state iterations will start. Now in every state control unit will pass the value that goes to shifter. for performing specific shift operation on input data. When all iterations gets completed at last state control unit will release ‘ena’ signal that will disable output registers. Further no feedback will be given. Final output is presented. As shown in the state diagram the states are followed.

VI VHDL Implementation

As described earlier basic CORDIC unit is implemented using add-sub function look-up table and sign control logic Rotation mode is executed for sin, cos calculation. The outer most entity will deal with i/p and o/p



.

fig 2 Entity sincos

As shown in fig reset and load is applied first clock and enable is set angle input is given. Since from the second state feedback is given to the input hence load is made ‘0’ unless and until 16 iterations are completed. Output is obtained at sin and cos port outputs respectively. The outermost entity is sincos_wordserial which will take angle

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

input and will give sine cos as output. The required components are instantiated in ths main module. Figures 1 shows the hardware for serial scheme ,shifting and feedback frome x,y,z is seen in RTL

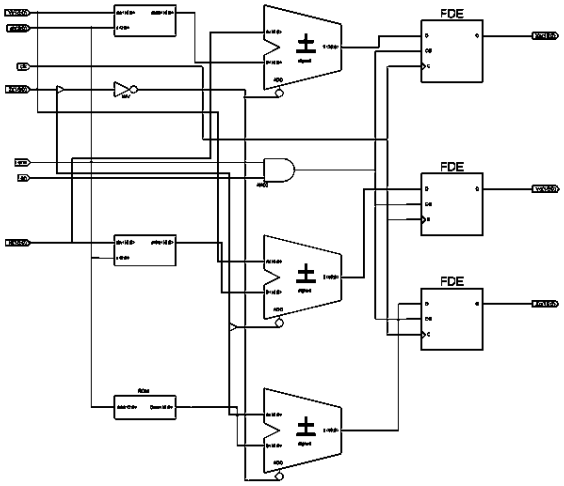


FIG 3 BASIC CORDIC UNIT

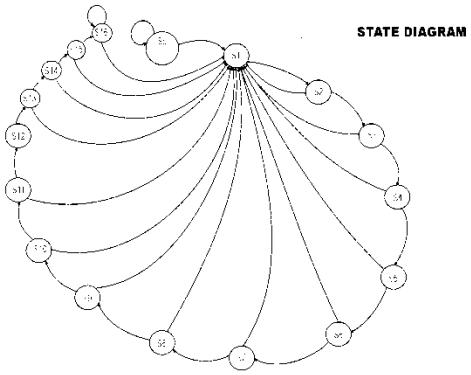


fig 4 state diagram

Simulation result

Simulation results are presented for each module for Word_Serial scheme For four specimen angle i/ps output results are shown in following figures

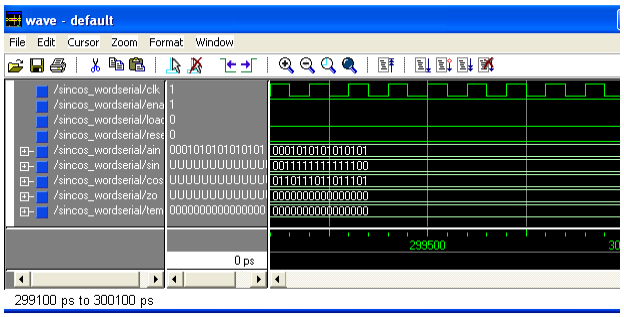


fig. 5 Simulation Result For 30^0

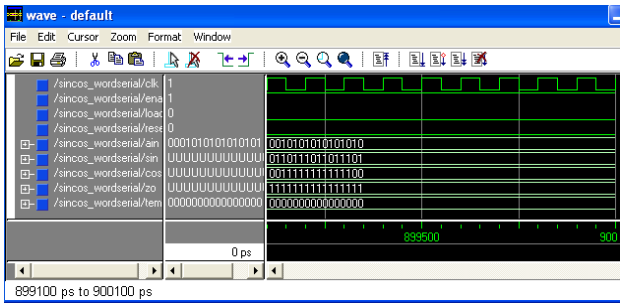


fig.6 Simulation Result For 60^0

V CONCLUSION

Various CORDIC architectures exist. Determining the selection of a proper architecture for a CORDIC processor is a complex task that requires the analysis of different design strategies and evaluation of the principal parameter, speed of implementation, area consumption and accuracy, in order to obtain the best design for a specific application.CORDIC algorithm is hardware efficient algorithm it fits in smaller area using iterative style. Every iteration takes one clock cycle so that in 'n' clock cycles, 'n' iterations are performed. configuration for CORDIC Word_Serial (iterative) is devised. The extra hardware required is control unit for this architecture. State machine is designed for this.

REFERENCES

1. Volder, J."The CORDIC Trigonometric computing technique", IRE Trans. Electronic computing, vol EC-8,pp330-334,sept 1959.
2. Andraka R."A survey of CORDIC Algorithms for FPGA based computers", proc. Of ACM/SIGDA Sixth International Symposium on FPGAS,Feb,1998,Monterrey,CA,pp-191-200
3. A VHDL Implementation of a CORDIC Arithmetic Processor chip faculty of computing technology technical report 94-9
4. Tradeoffs in Arithmetic Architectures for CORDIC Javier Oscar Giacomantone
5. Duprat,J.and Mullar J.M."The CORDIC Algorithm :New Results for Fast VLSI Implementation ,"IEEE transactions on computers,vol,42,pp,168-178,1993
6. Walther J.S."A Unified algorithm for elementary functions ", Spring Joint Computer conf. 1971 proc.pp-379-385
7. Hu,Y.H "The quantization effects of thec CORDIC algorithm ",IEEE Trans.on signal processing vol.40 no,4,april 1992,pp,834-844
8. Linhardt,R.J,and Miller H.S.,"Digit – by – Digit Transcendental function computations",RCA,Rev.30(1969) pp 209-247
9. Precomputation of Rotation Bits in Unidirectional CORDIC for Trignometr and Hyperbolic Computations,Satish Ravichandran, Vijay Asari old Dominion University,Norfolk,Virginia 23529,USA