OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Performance Analysis of CORDIC Architectures Targeted by FPGA Devices

Guddeti Nagarjuna Reddy[1], R.Jayalakshmi[2], Dr.K.Umapathy[3]
M.Tech (EDT), SCSVMV UNIVERSITY, Kanchipuram, India[1]
Asst.prof, SCSVMV UNIVERSITY, Kanchipuram, India[2]
Assoc.prof, SCSVMV UNIVERSITY, Kanchipuram, India[3]

ABSTRACT:- *A wide variety of elementary functions are evaluated by CORDIC algorithm. This paper provides the performance analysis of folded and unfolded CORDIC architectures at different parameters. It also provides the latency comparison for the two structures. Same synthesis description is used for all the structures. For word lengths of 16 and 32 bits it gives the maximum operating frequency comparison of folded, unfolded and pipelined structures.*

Index Terms:- *CORDIC algorithm, folded architecture, unfolded architecture.*

I. INTRODUCTION

The CORDIC means coordinate rotation digital computer. It was proposed by Volder in 1959. CORDIC is an efficient iterative algorithm which can be used to compute several elementary functions. Another scientist named Walther extended it to encompass circular, linear and hyperbolic coordinate systems. In a variety of applications the algorithm is commonly used to perform rotations and also to evaluate elementary functions. Long latency problem occurs.

Based on the hardware realization of the three iterative equations CORDIC architectures are broadly classified. They are folded and unfolded architectures. In time domain folded architectures are multiplexed, so that in a single processing architecture it provides a means for trading area. By using a word serial design the entire CORDIC core is implemented in folded architectures.

II. CORDIC ALGORITHM

By using only shift and add operations the CORDIC algorithm provides an iterative method of performing vector rotations by arbitrary angles. From general rotation transform the algorithm is derived.

$$xn = Bn.(x1 \cos z1)$$
$$yn = Bn.(x1 \sin z1)$$

In two modes CORDIC rotator is normally operated. In the rotation mode, with the desired rotation angle, the angle accumulator is initialized. The magnitude of the residual angle in the accumulator is made to diminish by rotation decision at each iteration.

2.1 FOLDED WORD SERIAL DESIGN

A iterative bit-parallel design, also called folded word design serial design. In hardware by simply duplicating each of the three different equations is obtained.

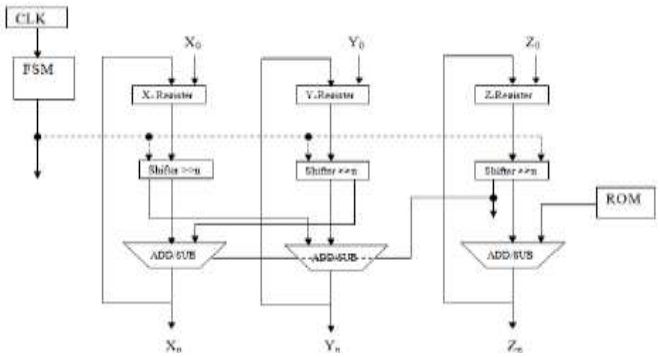

Fig. a folded word serial CORDIC

After each iteration being a shift- add algorithm, each individual unit consists of an adder or subtractor unit, the computed values are holding by a register and shifter. To start with, via a multiplexer the initial are fed into each branch. The operation of the adder-subtraction unit is determined by the value of *z* branch.

Through the shifter unit signals in the *x* and *y* branch are then added to or subtracted from the unshifted signal in the opposite path. According to the number of iteration, the *z* branch arithmetically combines the register values from looking up table, table whose address with values are taken is changed. Hence the result of this operation determines the nature of operation for the next iteration. Directly the results are read from the adder or subtraction units after *n* iterations. By tracking of shifting distances and the ROM addresses a finite state machine is used.

In high speed applications, the conventional approach of implementing adder or subtractor units and shifters on time basis at every path are shared. Another disadvantage is with respect to the shift operations. With the number of iteration the shifters have to change the shift distance when implemented in hardware.

A high fan in is required for large number of iteration and reduce the maximum speed for the application. Into the FPGA architectures these shifters do not map well and requires several layers of logic if implemented, results a slow design that uses large number of logic cells. In addition where *n* is the number of iterations the output rate is also limited by the fact that the operation is performed iteratively and therefore the clock state rate equals 1/*n* times the maximum output rate.

2.2 UNFOLDED PARALLEL DESIGN

As per the discussion of above demands that the processor has to perform iterations at *n* times the data rate, this is the iterative nature of the CORDIC processor. Each of *n* processing elements always performs the same iteration for unfolded iteration process. To design parallel processing architectures from serial processing architectures, a direct application of the unfolding transformation is used. The means of word-parallel architectures can be designed from word serial architectures at the word level.

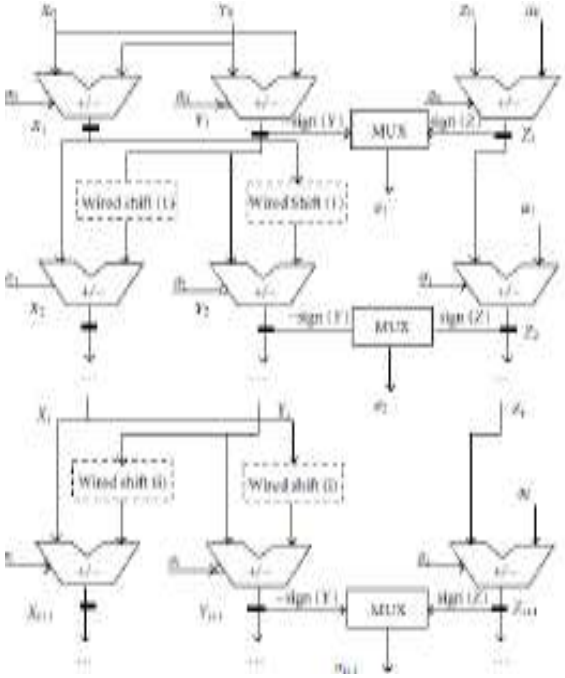

Fig. Unfolded CORDIC design

The CORDIC processor results in two significant changes in the unfolded architecture. First, it has to perform a constant shifting operation in each stage that is the shifter in each unit is of fixed shift.

As in the iterative structure the shifter needs not be updated, and makes the FPGAs implementation quite feasible. Second, during iteration from the process the unfolding process eliminates the use of ROM which was required to hold the constant angle values. Instead of requiring storage space those constants can be hardwired, reducing to an array of interconnected adder or subtraction units of entire CORDIC processor.

Making the unrolled processor strictly combinational, the need for registers is also eliminated. The processor can be easily pipelining by insertion of registers between the adder-subtraction units is another advantage of the unrolled design. In each logic cell, there are already registers are present in most FPGA architectures, so the addition of the pipelining registers has not addition hardware cost.

III. IMPLEMENTATION AND RESULTS

Methodology

The CORDIC processor is implemented in seven stages and for a word length of 16 and 32 bits. The initial design entry of both architectures is done using VHDL. The design translation is carried out in Xilinx ISE 12.4. The simulator database is then analyzed for both designs at different performance parameters and logical conclusions are drawn. Implementing the core with the following synnthesis:

Platform: Field Programmable Gate Array
Family: Virtex5
Target device: XC5VLX30
Package: FF324

3.1 ANALYSIS AND RESULTS

For different performance parameters folded and unfolded architectures are analyzed, and provides latency comparison. Same synthesis description is implemented for all structures.

TABLE 1 LATENCY COMPARISON FOR16 AND 32-BIT CORDIC

Parameter	CORDIC architectures.			
	Folded		Unfolded	
	16 Bit	32 Bit	16 Bit	32 Bit
Logic delay.	5.594ns	6.959ns	5.804 ns	9.18 ns
Route delay.	25.023ns	33.071ns	18.69 ns	25.0 ns
Max. Combinational delay.	33.078ns	42.414ns	24.472ns	34.2 ns

TABLE 2 THROUGHOUT COMPARISON FOR16 AND 32-BIT CORDIC

Parameter	CORDIC architectures.					
	Folded		Unfolded (parallel)		Unfolded (pipelined)	
	16 Bit	32 Bit	16 Bit	32 Bit	16 Bit	32 Bit
Max. operating frequency	216.57MHz	125.27 MHz	44.29 MHz	31.67 MHz	232.6 MHz	163.43 MHz

It gives the maximum operating frequency comparison of the folded, unfolded and pipelined structures for word lengths of 16 and 32 bits.

It is observed that when timing response of the CORDIC structures is concerned, the unfolded architecture has less worst-case delay compared to the folded structure. This is due to the unfolding process which eliminates the use of registers and therefore the corresponding setting-up and hold times. The overall latency is thus reduced by a factor proportional to those setting-up and holding times. Note, however that the maximum operating frequency and thus the throughput of the unfolded CORDIC design is determined by the worst case delay of the structure. This is because the structure is pure combinatorially. Contrast to this, the folded structure can be clocked at high frequencies resulting in large operating frequencies. However, pipelining the unfolded CORDIC makes it possible to process multiple inputs simultaneously, there by increasing the maximum operating frequency of the unfolded structures. At an N stage CORDIC core, N stage pipeline can give maximum results. At first output of an N-stage pipelined CORDIC core is obtained after N clock cycles. Therefore, outputs will be generated after each every clock cycle. Further analysis of CORDIC is carried out by comparing the power consumption for 32 bit word length.

It gives the power consumption for the three structures.

Folded structures have less power dissipation compared to the parallel and pipelined structures. The power consumed by logical components in case of folded structures is quite low. This is due to the fact that the folded structure uses the same components repetitively. Similarly due to the multiple input/output instantiations in unfolded structures the power consumed by the input and output resources is quite high resulting in high dynamic power dissipation in the parallel and pipelined designs. Finally the three designs are analyzed for area consumption in terms of resource utilization and the results are tabulated.

TABLE 3 POWER COMPARISON FOR 32-BIT CORDIC

Instance (resource)	CORDIC architectures.		
	Folded	Unfolded (parallel)	Folded (pipelined)
power (clock)	21.32 mW	--	17.75 mW
power (logic)	2.15 mW	13.87 mW	9.20 mW
power (signals)	15.71 mW	11.01 mW	12.33 mW
power (IOs)	93.60 mW	196.07 mW	196.64 mW
power (leakage)/ quiescent	380.99 mW	382.30 mW	382.21 mW
dynamic power	132.78 mW	220.95mW	235.92 mW
total power dissipation	513.77 mW	603.25 mW	618.13 mW

As expected, the folded structure is an efficient user of logic since the same logical units are used for every iteration. But since the results need to be fed back after for every iteration a large number of registers are used in the folded word serial implementation.

TABLE 4 AREA COMPARISON FOR 32-BIT CORDIC

parameter	CORDIC architectures.		
	Folded	Unfolded (parallel)	Folded (pipelined)
No. of Registers	768	--	678
No. of LUTs	287	1093	1006
No. of logic blocks used	285	1093	1006
No. of occupied Slices	121	589	336
No. of LUT Flip Flop pairs used	768	1093	1013
No. of bonded IOBs	193	193	194

IV. CONCLUSIONS

In this paper we proposed a simple operation by using VHDL coding in FPGA platform of Virtex5 family. From this we concluded analyzed the area, power consumption, layout comparison and throughout comparison for folded, unfolded and folded pipelined CORDIC architectures.

REFERENCES

[1]. J.E VOLDER, "The CORDIC Trignometric Computing Technique" IRE Tran Electronic Computers vol EC-8 PP 330-334,1959

[2]. J.S. Walther," A Unified Algorithm for Elementary Functions, " proc., Spring Joint Computer Conf., Volume 38.,pp., 379-385, 1971.

[3]. S. Wang, V. Piuri, and E.E. Swartzlander, Jr, " Hybrid CORDIC Algorithms, " IEEE Transaction Computers, Volume 46, Number 11., pp., 1202, November 1997.

[4]. D.S. Phatak, " Double Step Branching CORDIC : A New Algorithm for Fast Sine and Cosine Generation, IEEE Transaction Computers, Volume 47 Number 5., pp., 587-602, May 1998.