

Design Issues in Synthesis of Reusable Cores

Rohit Sharma
Texas Instruments (India) Ltd.
rsharma@ti.com

C.P. Ravikumar
Indian Institute of Technology, Delhi
rkumar@ee.iitd.ernet.in

Abstract

While core-based design is itself a challenging task, it is equally challenging for a core vendor to provide information about a core without compromising on the protection of intellectual property. A number of issues are to be taken into consideration when designing a core. While conventional goals such as minimal area and maximal performance continue to hold, additional constraints such as core testability and power dissipation will have to be considered. Since the vendor of a core does not reveal details about the internals of the core, it is often the responsibility of the vendor to provide the test plan for the core. In this paper, we present our experiences in designing a testable CORDIC core.

Keywords: Embedded Cores, Design Reuse, CORDIC Arithmetic, and Core Testability.

1. Introduction

A number of ASIC vendors are making their designs (intellectual property) available to designers in the form of reusable cores. A core may be a soft core or a hard core, depending on the amount of detail made available to the designer by the vendor. A system-on-chip designer integrates hard/soft cores into the target design, in a manner similar to off-the-shelf chips being integrated into a system design. The difference, however, is that off-the-shelf chips can be separately tested. Cores, on the other hand, become an integral part of a larger chip and have to be tested as part of the complete system. Since the designer does not know the internals of the core, developing the tests for the core is the responsibility of the core vendor. Making details such as scan path(s) available to the outside world may compromise the protection of vendor's intellectual property. At the same time, alternate test techniques may not provide the desired fault coverage.

In this paper, our aim is to study the testability aspects in the design of a reusable core. We consider the example of the CORDIC (COordinate Rotation DIgital Computer)

core for computing trigonometric functions [Volder, 1959]. Section 2 describes the design of the CORDIC core itself. In Section 3, we highlight the various design considerations for embedded cores. Section 4 describes the test-related issues in the design of the core. Experimental results are provided in Section 5. Conclusions are presented in Section 6.

2. CORDIC Core

The CORDIC technique for computing trigonometric functions was first given by Jack Volder [Volder 1959]. Several variations of CORDIC have been proposed since then. Some significant milestones in CORDIC-related research are the following.

- The CORDIC algorithm has been adapted for the computation of hyperbolic trigonometric functions, inverse trigonometric functions, exponential and logarithmic functions [Hayes 1989]
- Takagi et al. presented a technique to speed up the CORDIC iterations [Takagi 1991].
- Hu and Naganathan [Hu 1993] showed that when the angle θ is known apriori, it is possible to recode the angle and express it as the sum of very few elementary angles of rotation. Hu and Chern [Hu 1996] used a similar angle recoding technique to improve the efficiency of a backward rotation CORDIC algorithm.

Since the CORDIC algorithm does not use multiplications, it is especially well suited for VLSI implementation. Thus, CORDIC "cells" can be used in VLSI realizations of digital signal processing algorithms such as Fast Fourier Transform [Sarmiento 1998] and Singular Value Decomposition [Ercegovac 1990].

2.1 CORDIC Algorithm

We shall describe only the pseudo code of original CORDIC algorithm given by Volder [Volder 1959] to compute the sine and cosine functions of a given angle θ . The details have been omitted in the interest of brevity.

procedure CORDIC (θ)

```

begin
  X := 1/K;
  Y := 0; // Vector of size 1/K along X axis
   $\phi = 0$ ; //  $\theta - \phi$ : angle by which vector must be rotated
  for j := 0 to n-1 do
    begin
      if ( $\phi < \theta$ ) then
        begin // Rotate anti clockwise
           $X := X - Y/2^j$ ;
          //Division realized through right shift
           $Y := Y + X/2^j$ ;
           $\phi = \phi + \tan^{-1}(1/2^j)$ ;
          // Access prestored angle thru lookup
        end
      else
        begin // Rotate clockwise
           $X := X + Y/2^j$ ;
           $Y := Y - X/2^j$ ;
           $\phi = \phi + \tan^{-1}(1/2^j)$ ;
        end;
      end;
    end;
  end;
end;

```

2.2 CORDIC Architecture

The pseudo code presented in Section 2.1 reveals that the main computations in the CORDIC algorithm are addition, subtraction, shifting, and table lookup. To maximize the performance of the CORDIC design, it is necessary to implement a fast addition/subtraction algorithm, a fast shift operation, and a fast table lookup. Barrel shifters are normally used in CORDIC implementations. Table lookup is implemented as a combinational circuit (PLA). Three types of adders have been implemented, ripple carry adder (RCA), carry lookahead adder (CLA) and redundant binary adder (RBA) with time complexity of $O(n)$, $O(\log n)$ and $O(1)$ respectively. The use of an RBA calls for an alternate ("redundant binary") number representation, where the base is *ternary* and the digits come from the set $\{-1, 0, 1\}$. It is possible to avoid carry propagation during the addition of two "redundant binary" numbers (For details, see [Ravikumar 1995], [John 1998], and [Sharma 1998]). Figure 1 shows the basic architecture of a CORDIC cell. The width of the data path is n , when we employ n -bit accuracy and use either RCA or CLA for addition. Several changes is required in the data path when we make use of redundant binary addition, which results in more area consumption. Nevertheless, a "redundant" CORDIC algorithm is expected to be faster. We implemented four different data paths (12 bit wide) and control paths for a CORDIC cell. These differ in the type of adder used viz. 12-bit RCA, 12-bit CLA, 4 bit CLA

(Three 4-bit CLAs and carry ripple across stages of CLA) and 12-bit RBA.

3. Design Issues

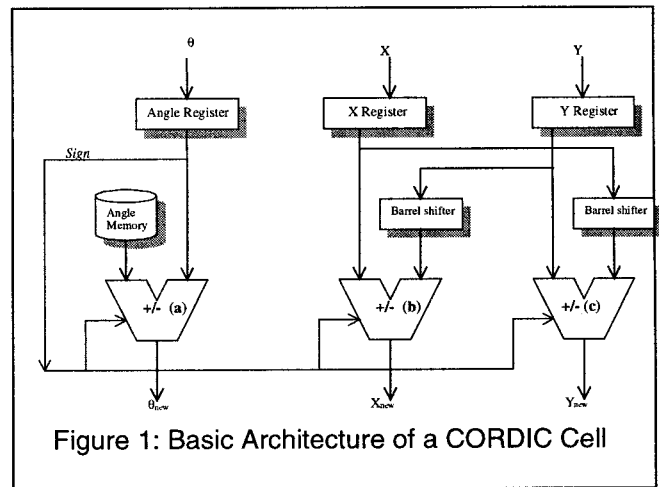


Figure 1: Basic Architecture of a CORDIC Cell

The data path architecture plays a crucial role in determining the area and performance of the final design. Other than area and performance, average and peak power dissipation become important considerations in DSP applications that run on battery power e.g. CORDIC cells used in a robotics application. Testability of the core is measured by several factors such as the number of test vectors required to obtain specified fault coverage, the additional area overhead and the loss in performance incurred for testability purposes.

The data path architecture, the selected fabrication technology, the tools used to compile the design will all influence the area, performance, testability, and power dissipation of the design core. The system designer has to perform a careful evaluation of the above design metrics before selecting the component (core) for integration into the system.

4. Testability Issues

Functional design units such as the CORDIC cell, FFT cell, DCT cell, MPEG encoder, and so on, are often reusable in system-on-chip designs. However, core-based design has several challenges. In order to test the system after fabrication, it is necessary to test the internals of the core, but the system designer has no knowledge of the internals. Vendor of the core has to take the responsibility of providing the test methodology for the core. At the same time, the requirements of the design, as far as the system testing is concerned, may be dictated by the overall system-level area/performance tradeoffs. Therefore, a system designer will usually look for a

variety of options such as Built-in Self Test, Scan Path testing, Partial Scan, and other mixed solutions for testing the cores. For the CORDIC cell considered in the previous section, we shall describe several test solutions and compare them on a relative basis.

4.1 Scan Path Testing

Scan testing may be based either on *full scan* or *partial scan*, depending on the number of flip-flops included in the scan chain. The disadvantages of full scan testing are large area overhead, performance degradation and compromise on the security of the design. A partial scan solution reduces the area and performance overhead by selecting some of the flip-flops for inclusion in the scan path. The price to be paid is a reduction in fault-coverage and an increase in the number of test patterns. In Section 5.1, we report the results of applying full and partial scan test techniques to the various CORDIC designs.

4.2 Built-in Self Test

Built-in Self-Test (BIST) relies on random test pattern generation and signature-based testing of the circuit under test. Examining the CORDIC architecture of Figure 1, we see that the data path can be tested in several ways. We can transform the X and Y registers into pseudo-random pattern generators and configure the barrel shifters into identity function generators (shift by 0) during test mode. Two extra registers X_{new} and Y_{new} will have to be added to compact the responses of adders (b) and (c) into signatures. Similarly angle register will require two additional register for parallel testing. The above BIST solution will allow all the adders to be concurrently tested. However, this solution will require a large area overhead. Since all the adders are simultaneously tested using random vectors, the power dissipation during testing can be expected to be high. Alternate BIST architectures, for instance sequential test plan, can alleviate area overhead and test power dissipation problem.

4.3 JTAG Boundary Scan

The IEEE Standard 1149.1 JTAG boundary scan [Texas Instruments 1997] inserts one boundary scan cell (BSC) per function pin of the core. Additional test interface logic, called test access port (TAP) is required to make a core boundary scan testable. We explored the option of using JTAG boundary scan along with full internal scan to add testability to the CORDIC cell. The area overheads of the TAP interface and full scan path were significant in comparison to the area of the core (see Section 5.)

5. Results

We used two synthesis-based design flows in order to generate the various designs for the CORDIC core. The SynergyTM synthesis tool from Cadence was used with a 0.8 μ , 3-metal SCMOS technology library (borrowed from the Engineering Research Center of Mississippi State University) to compile descriptions of CORDIC designs written in the Verilog HDL. The Synopsys *Design Compiler*TM (DC) was employed along with a 0.18 μ , 5-metal CMOS library (TGC6000, Texas Instruments) to compile the Verilog HDL CORDIC designs. The Synopsys *Test Compiler*TM was used to generate the test interfaces and the test vectors for all the designs. Table 1 shows that the architecture based on 12-bit RBA gives the best performance, but occupies the largest area. The architecture based on 12-bit RCA has the least area as well as least performance. Table 2 shows the results obtained using the 0.18 μ technology library and DC.

Table 1 : CORDIC designs using 0.8 μ m technology and Synergy Synthesizer

Arch.	Area (λ^2)	Library Cell Count	X-sistor Count	Max. Speed (MHz)	Critical Path Delay (ns)
RCA	45192	1057	8493	46.85	21.35
CLA1 ¹	48169	1280	8955	52.83	18.93
CLA2 ²	51241	1441	9759	57.44	17.44
RBA	78312	1750	12096	76.33	13.10

5.1 Testability Results

43 flip flops were used for RCA or CLA based CORDIC cells while RBA based CORDIC cells uses 77 flip flops with full scan solution. The area overhead of using full scan (along with JTAG boundary scan) for the RCA CORDIC cell is about 67%. The figure for the CLA and RBA based CORDIC is 53% and 35.9% respectively. Comparing Table2 and Table 4 obviates the small performance degradation due to the insertion of boundary scan and full scan. Critical path delay of RBA based CORDIC cells (fastest among all cells) from 5.9 ns to 6.02 ns, an increase of 2%. The test solutions are applicable when arrays of CORDIC cells are used in a larger DSP application. Table 3 shows the results of the testability study on the CORDIC cores. 12-bit CLA based architecture is best in terms of test patterns and fault coverage. Very small controllability of the Test Data Input (TDI) pin over the additionally inserted JTAG components (e.g. TAP controller and Instruction Register) is cause of significant fall in fault coverage.

¹ 12 bit CLA based CORDIC design.

² 4 bit CLA based CORDIC design.

Table 2: CORDIC Core using 0.18 μm and DC

Arch.	Area (λ^2)	Max. Delay (ns)	Speed (MHz)	Power (mW) Cell Internal	Power (mW) Net Switching
RCA	1034	11.09	90.18	2.13	1.53
CLA1	1302	9.89	101.11	2.79	2.5
CLA2	1904	7.94	125.90	4.30	3.97
RBA	3210	5.9	169.50	8.52	8.54

Table 3: Testability Results for the CORDIC Core³

Arch.	With Full Scan	JTAG with Full Scan	With Partial Scan
RCA	70 TP 43 SC 99.92% FC	71 TP 43 SC 75.9% FC	95 TP 39 SC 88.95% FC
CLA	56 TP 43 SC 100% FC	56 TP 43 SC 80.39% FC	100 TP 42 SC 96.73% FC
RBA	103 TP 77 SC 100% FC	101 TP 77 SC 81.88% FC	191 TP 71 SC 90.71% FC

Table 4: CORDIC Core with JTAG and scan-chains using 0.18 μm library and DC.

Arch.	Area (λ^2)	Max. Delay (ns)	Speed (MHz)	Power (mW) Cell Internal	Power (mW) Net Switching
RCA	1727	11.35	88.10	2.81	2.05
CLA1	1997	10.02	98.81	3.67	3.15
CLA2	2602	8.09	123.61	5.68	5.3
RBA	4363	6.02	166.11	11.72	11.77

6. Conclusions

In this paper, we have studied the design and testability issues that arise in the automatic synthesis of embedded cores. We used the CORDIC computer as a case study. We have seen that the data path architecture, the fabrication technology, and the synthesis tools influence the area, performance, power, and testability characteristics of the core. Our results for the CORDIC computer indicate that no particular architecture can be

declared as superior to others in all respects. The system designer who integrates the cores into a larger design must carefully consider the various architectural options available and select the appropriate component. Due to the same reason, the core vendor must make several design variations available to the system designer. We have seen that it is difficult to provide all the design characteristics uniformly for all the designs that we have reported in this paper. This makes it difficult for the system designer to take design decisions. Extrapolation and statistical techniques will become inevitable in such situations to obtain accurate guesses of missing information.

References

- [Ercegovac 1990] M.D. Ercegovac and T. Lang. *Redundant and On-line CORDIC, Applications to Matrix Triangularization and Singular Value Decomposition*. IEEE Transactions on Computers, Vol. 39, No. 6, June 1990.
- [Hayes 1989] J.P. Hayes. *Computer Organization and Architecture*. McGraw Hill Publishers. Second Edition, 1989.
- [Hu 1993] Y.H.Hu and S.Naganathan. *An Angle Recoding Method for CORDIC Algorithm Implementation*. IEEE Transactions on Computers. Vol. 42, No. 1, pp. 99-101, January 1993.
- [Hu 1996] Y.H.Hu and H.H.M.Chern. *A Novel Implementation of CORDIC Algorithm Using Backward Angle Recoding (BAR)*. IEEE Transactions on Computers. Vol. 45, No. 12, pp. 1370-1378, December 1996.
- [John 1998] B. John. *Low Power CORDIC Chip*. M.Tech Thesis. Department of Electrical Engineering, Indian Institute of Technology, Delhi. January 1998.
- [Ravikumar 1995] C.P. Ravikumar. *A Redundant CORDIC Chip*. Proceedings of SIGCOM 95, Bangalore 1995.
- [Sarmiento, 1998] Roberto Sarmiento et al, *A CORDIC Processor for FFT Computation and Its Implementation Using Gallium Arsenide Technology*. IEEE Transactions on VLSI Systems, Vol. 6, No. 1, March 1998.
- [Sharma 1998] R. Sharma. *Cell Based and FPGA Based ASIC Design of CORDIC*. M.Tech Thesis. Department of Electrical Engineering, Indian Institute of Technology, Delhi, January 1998.
- [Takagi 1991] N. Takagi, T.Asada, S.Yajima. *"Redundant CORDIC Methods with a Constant Scale Factor for Sine and Cosine Computation."* IEEE Transactions on Computers. Vol. 40, No. 9, pp. 989-995, September 1991.
- [Texas Instruments 1997] Semiconductor Group, Texas Instruments. IEEE Std 1149.1 (JTAG) Testability Primer. 1997.
- [Volder 1959] J.E.Volder. *"The CORDIC Trigonometric Computing Technique,"* IRE Transactions on Electronic Computers, Vol. 8, No. 3, pp. 330-334, 1959.

³ TP: Test Patterns, SC: Scan cells, FC: Fault Coverage.