ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.10, No.4, 1–8

A high-throughput fixed-point complex divider for FPGAs

Dong Wang[1a)], **Pengju Ren**[2b)], **and Leibo Liu**[3c)]

[1] *Institute of Microelectronics, Tsinghua University*

[2] *Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University*

[3] *Research Center for Mobile Computing, Tsinghua University*

a) *doonny1212@gmail.com*

b) *pengjuren@gmail.com*

c) *liulb@tsinghua.edu.cn (Corresponding author)*

Abstract: This paper presents a high-throughput fixed-point complex divider which uses four pipelined CORDIC units to transform and divide complex numbers in Polar coordinates. By persevering the macro-angle for CORDIC rotations in redundant form and developing an optimized pipelining structure, the FPGA based implementation achieves a 9× advantage on throughput over the best design reported. In addition, the final error is guaranteed within 1 ulp (unit in last position). Thus the proposed complex divider is highly suitable for accelerating DSP applications with high precision numerical accuracy requirements.

Keywords: complex divider, FPGA, CORDIC

Classification: Electron devices, circuits, and systems

References

[1] F. Edman and V. Oewall, "Fixed-point implementation of a robust complex valued divider architecture," *Proc. European Conf. Circuit Theory and Design*, Cork, Ireland, pp. 143–146, Aug. 2005.

[2] R. L. Smith, "Algorithm 116: complex division," *Communications of the ACM*, vol. 5, no. 8, p. 435, 1962.

[3] J. Liu, B. Weaver, and Y. Zakharov, "FPGA implementation of multiplication-free complex division," *Electronic Letters*, vol. 44, no. 2, pp. 95–96, Jan. 2008.

[4] P. Dormiani, M. D. Ercegovac, and J.-M. Muller, "Design and implementation of a radix-4 complex division unit with prescaling," *IEEE Int. Conf. Application-Specific Systems and Architectures*, Los Angeles, USA, pp. 83–90, April 2009.

[5] D. Wang, M. D. Ercegovac, and N. Zheng, "A Radix-8 complex divider For FPGA implementations," *19th Int. Conf. Field Programmable Logic and Applications*, Prague, pp. 236–241, Aug. 2009.

[6] M. D. Ercegovac and J.-M. Muller, "Complex division with prescaling of operands," *IEEE Conf. Application-Specific Systems and Architectures*, Los Angeles, USA, pp. 304–314, June 2003.

[7] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron Compute*, vol. 8, no. 3, pp. 330–334, 1959.

DOI: 10.1587/elex.10.20120879
Received November 20, 2012
Accepted January 21, 2013
Published February 19, 2013

[8] R. Andraka, "A survey of CORDIC algorithms for FPGA based computers," *Proc. the Sixth ACM/SIGDA Int. Symp. Field Programmable Gate Arrays*, pp. 191–200, 1998.
[9] K. Kota and J. R. Cavallaro, "Numerical accuracy and hardware tradeoffs for CORDIC arithmetic for special-purpose processors," *IEEE Trans. Comput.*, vol. 42, no. 7, pp. 769–779, 1993.
[10] D. Lewis, et al., "The Stratix-II logic and routing architecture," *Int. Symp. Field Programmable Gate Arrays*, pp. 14–20, 2005.

1 Introduction

Complex division is a fundamental operation in Digital Signal Processing (DSP) and numerical computation applications such as Vertical Bell Laboratories layered Space-Time (V-Blast) detection, Orthogonal Frequency-Division Multiplexing (OFDM) and channel equalization of MIMO systems. Calculating complex division is a complicated task especially when hardware implementations, such as FPGA based designs, are considered. Achieving low computation latency and high throughput are the two major concerns on performance when exploring the hardware architecture. Recently, several designs, adopting various algorithms, have been presented in the literature to address these two issues.

In [1], a complex divider is described, simply implementing the Smith's algorithm [2] with a group of pipelined real numbers multiplier, divider, and adder on the Xilinx Virtex-II FPGA. The implemented design operates at 100 MHz with an average throughput of 28 MOPS (Mega Operations Per Second) at the cost of 117 logic slices. The maximum precision achieved is only 8-bit (with +4 guard bits). Since the Smith's algorithm does not always guarantee an exact division result when implemented in fixed-point format, this approach will inevitably affect the accuracy of the target application, such as reducing the Signal-to-Noise Ratio (SNR).

In [3], a multiplication-free complex division unit is proposed, which implements the Dichotomous Coordinate Descent (DCD) algorithm for 15-bit (Q15 format) computation. The implemented unit on the Xilinx Virtex-II XC2VP30 FPGA costs 527 logic slices. Although the DCD algorithm only require bit-shift and addition operations, the obtained average throughput is only 1.6 MOPS under a maximum clock frequency of 100 MHz due to the large number of iterations needed to be performed.

Later, two high-radix complex divider designs with reduced latency and improved accuracy are reported in [4] (a radix-4 design) and [5] (a radix-8 design), respectively. These two designs are based on the digit-recurrence algorithm proposed by Ercegovac et al. [6], which utilizes the operands prescaling technique to simplify the selection of complex quotient digits. The radix-4 design uses the embedded DSP blocks on Altera Stratix-II FPGA to accelerate the multiplications in the prescaling step. By decomposing the digit 3 into $(2+1)$, the digit-recurrences are simply performed by a group of hand-

DOI: 10.1587/elex.10.20120879
Received November 20, 2012
Accepted January 21, 2013
Published February 19, 2013

optimized redundant adders. The implemented 16-bit unit costs 566 LUTs, 8 DSP blocks and a prescaling look-up table of 2 K words of 16 bits. The corresponding throughput is 9.2 MOPS under the clock frequency of 175.9 MHz.

The radix-8 design utilizes a higher radix to further reduce the computation latency, resulting in a more complex digit-set of $\{-7, -6, \cdots, +6, +7\}$. Therefore, it proposes to share a group of four multipliers between the prescaling and recurrence computations to optimize the hardware costs by using a modified Booth encoder. The final design is implemented on Altera Stratix-II S60F672C3 FPGA and the maixmum throughput achieved is 10.6 MOPS under the frequency of 95.4 MHz for 16-bit computation. The advantage of the high-radix digit-recurrence based approaches is that the computation latencies could be greatly reduced, i.e., 19 cycles (radix-4) and 9 cycles (radix-8) for 16-bit complex division. However, unfolding the recurrences to implement a pipelined structure targeting high computation throughput is not economic, since both the prescaling and each of the iteration stages require four large multipliers.

In this paper, a high-throughput architecture of a fixed-point complex divider is proposed. Four CORDIC units with optimized structures are used to perform pipelined complex division operations in Polar coordinates. The design space is quantitatively explored and compared, showing a result of a reasonable trade-offs between area and performance. The 16-bit design is implemented in Verilog HDL and mapped into Altera Stratix-II FPGA, achieving a maximum throughput of 113.3 MOPS showing a competitive result over previous studies.

2 Performing complex division in Polar coordinates

Given two complex operands $a + bi$ (dividend) and $c + di$ (divisor), where $i = \sqrt{-1}$, the complex division is commonly defined as $(a + bi)/(c + di)$. Instead of directly performing $\left[(ac + bd)/(c^2 + d^2) + i(bc - ad)/(c^2 + d^2)\right]$ in Cartesian coordinates, the complex division could also be realized by using the transformed division formula:

$$p = (a\cos\theta + b\sin\theta)/\sqrt{c^2 + d^2} = \frac{a}{r}\cos\theta + \frac{b}{r}\sin\theta \tag{1}$$

$$q = (b\cos\theta - a\sin\theta)/\sqrt{c^2 + d^2} = \frac{b}{r}\cos\theta - \frac{a}{r}\sin\theta \tag{2}$$

where p and q denote the real and imaginary parts of the quotient. We also introduce two new variables $r = \sqrt{c^2 + d^2}$ and $\theta = \tan^{-1}(d/c)$ to represent the radial and the angular coordinates of the complex divisor $c + di$ in Polar coordinations.

In this paper, we propose to use four hardware CORDIC units [7] to implement the transformed division formula of (1) and (2) as Fig. 1 shows. The CORDIC unit (CORDIC-I) in the first stage works in the circular vectoring mode [8] and executes the transformation of the Cartesian coordinates (c, d) into Polar coordinates (r, θ). In the second stage, another two CORDIC units (CORDIC-II and -III) works in parallel, dividing the real and imaginary parts of the dividend $a + bi$ by the radius r. Both units are operating

DOI: 10.1587/elex.10.20120879
Received November 20, 2012
Accepted January 21, 2013
Published February 19, 2013

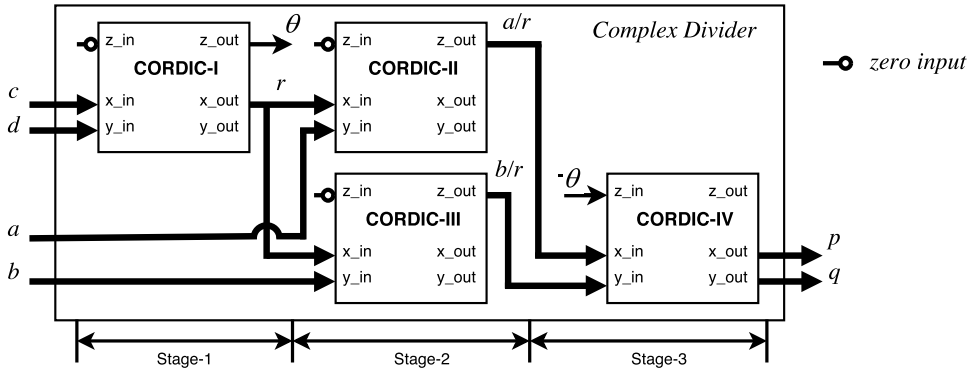

Fig. 1. Proposed complex divider.

in linear vectoring mode performing real divisions similar to a radix-2 non-restoring division algorithm. In the last stage, a circular rotation CORDIC unit (CORDIC-IV) rotates the immediate results a/r and b/r back with an angle of $-\theta$ to finish the complex division.

3 Error bounds and design parameters

In the proposed approach, multiple CORDIC units are concatenated sequentially, and thus the quantization error caused by fixed-point CORDIC computation [9] in the current stage will be propagated to the following stages and accumulated in the next CORDIC calculation. Therefore, to guarantee that the final error is within 1 ulp (faithfully rounded results), each of the intermediate CORDIC calculations should be correctly error bounded. By denoting the maximum quantization error of CORDIC-I as e_{S1} and the corresponding propagated value at the outputs of CORDIC-II as e_{P1}, we have

$$e_{P1} = \left|\frac{a}{r+e_{S1}} - \frac{a}{r}\right| = \left|\frac{a\cdot e_{S1}}{r(r+e_{S1})}\right| < \left|\frac{a}{r^2}\right| e_{S1} < \left|\frac{2}{K^2}\right| e_{S1} < e_{S1} \tag{3}$$

where $K \approx 1.64676$ represents a constant scaling factor introduced by the circular rotating CORDIC operation. Similarly, in the second stage, the quantization error e_{S2} and the propagated error e_{P2} satisfy

$$\begin{aligned} e_{P2} &= \left|\left[(\frac{a}{r}+e_{S2})\cos\theta - (\frac{b}{r}+e_{S2})\sin\theta\right] - \left[\frac{a}{r}\cos\theta - \frac{b}{r}\sin\theta\right]\right| \\ &= |e_{S2}\cos\theta - e_{S2}\sin\theta| < \sqrt{2}|\sin(\theta-\pi/2)|e_{S2} < \sqrt{2}e_{S2} \end{aligned} \tag{4}$$

Since the CORDIC algorithm only evolves shifting and adding operations, the propagated error are additive. Consequently, the total error at the outputs of CORDIC-IV is bounded by

$$e_{total} < e_{S3} + \sqrt{2}(e_{S2}+e_{S1}) \tag{5}$$

where e_{S3} denotes the quantization error of CORDIC-IV.

If a specific precision is desired in the final results, for instance, $e_{total} < 2^{-n}$ (n fractional bits), from (5), one could deduce that tighter bounds should be constrained as $e_{S3} < 2^{-(n+1)}$, $e_{S2} < 2^{-(n+3)}$ and $e_{S1} < 2^{-(n+3)}$, instead

DOI: 10.1587/elex.10.20120879
Received November 20, 2012
Accepted January 21, 2013
Published February 19, 2013

Table I. Summary of the design parameters for the four CORDIC units ($n = 16$).

CORDIC Number	Width of x/y channel (bit)			Width of z channel (bit)			Rotation Stage M
	Sign	Integer	Frac.	Sign	Integer	Frac.	
I	1	3	25	1	1	24	23
II/III	1	3	25	1	2	25	22
IV	1	2	24	1	1	27	21

of using the same bound 2^{-n} for all units. By using (5) and the quantization error bound formulas provided in [9], we have calculated the optimal parameters for each CORDIC units to implement a 16-bit fixed-point complex divider of the proposed architecture, in which the input operands are defined in the range of $1 \leq \max\{a, b\} < 2$ and $1 \leq \max\{c, d\} < 2$ as in [4] and [5]. The results are reported in Table. I. These parameters will be used in the following section.

4 Optimizing the pipelined CORDIC units

In this paper, we also present an optimized CORDIC structure, as illustrated in Fig. 2, to improve the computation throughput with minimum hardware cost. In order to avoid using separated adders to implement the addition and substraction in the CORDIC iterations [8], two bitwise inverters (the INVs) are utilized to obtain the opposite values of $x(i)$ and $y(i)$ (in 2's complement format) in the x/y channels. The bit-shifting operations are simplified as direct wiring and the MSBs are duplicated to implement sign-extension for substraction. The shifted results are swaped and input to the 29-bit fixed-point adder (for CORDIC-II/III and IV the adders are of 29-bit and 27-bit, respectively) with the direction control signal $sel(i)$ as the carry at LSB. In Altera Stratix-II FPGAs, the three-input addition can be efficiently mapped into a ternary adder structure [10] with hard-wired carry chain to speed up the propagation of the carry signal.

To further reduce the hardware costs, in CORDIC-I, the macro-angle $z(i)$ is kept in redundant form, i.e. only the sign bit, which is equivalent to $sel(i)$, is stored. Since the magnitude of the macro-angle $z(i)$ is predetermined, the back-rotation process in CORDIC-IV is carried out by sequentially adopting $sel(i)$ obtained from CORDIC-I. By using this optimization technique, twenty-three 26-bit adders are saved in the z channel resulting in a 26% area reduction for CORDIC-I. Similarly, in CORDIC-II/III, twenty-two 29-bit adders are avoided in the x channel since $x(i)$ remains the same during the CORDIC rotations.

Pipelining registers are inserted after the adders of every N-th unrolled iteration stage in the x/y channels. Therefore, the critical path is divided into $\lceil M/N \rceil$ pipelining stages, where M is the number of total rotation stages reported in Table. I. In CORDIC-I, $(2\lceil M/N \rceil + 2)$ 29-bit registers for x/y channels are utilized and $\lceil M/N \rceil$ 23-bit registers are used to pipeline the redundant angles in z channel as depicted in Fig. 2. Theoretically, a small

DOI: 10.1587/elex.10.20120879
Received November 20, 2012
Accepted January 21, 2013
Published February 19, 2013

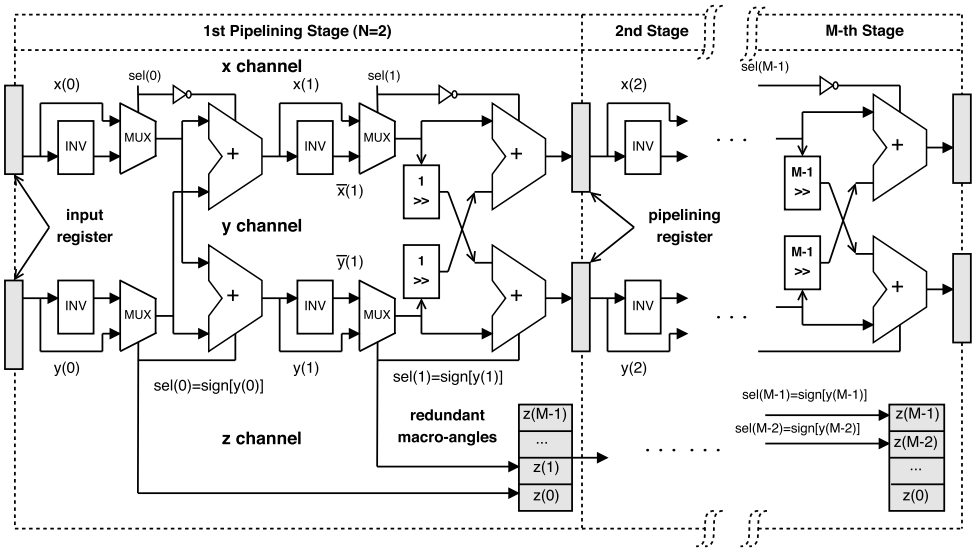

Fig. 2. Optimized CORDIC structure (CORDIC-I).

value of N results in a reduced delay on the critical path but a raise in the hardware costs. We will present the optimal value of N in the following section.

5 Results and comparisons

The proposed architecture was first modeled in Matlab programs by using the Fixed-Point Toolbox. Bit-accurate simulations were performed using more than 100 million randomly generated test vectors. The maximum computation errors were recorded and verified smaller than the theoretical bound. Afterwards, the design was described in Verilog HDL, simulated by Cadence NC-sim 5.1, synthesized and fitted into the Altera Stratix-II S60F672C3 FPGA using Altera Quartus-II 10.10 tool chain.

As discussed in previous section, the Verilog coded CORDIC units are parameterized with the variable N and implemented with the value ranging from $N = 1$ to $N = 12$. We report the area and performance characterization data for CORDIC-I in Fig. 3 and Fig 4. It is observed that starting from $N = 4$, the total area, especially the area of the registers, increases more dramatically as N decreases. In addition, the performance for $N = 4$ is 123 MHz, which is adequate for most applications. Therefore, we select $N = 4$ as the optimal parameter for CORDIC-I with balanced performance and cost. The architectural evaluation for other units are similar. The refined CORDIC units are then concatenated together to implement the proposed divider. The area and performance of the final design are reported and compared with four reference designs in Table II.

The proposed architecture achieves a maximum operating frequency of 113.3 MHz with a hardware cost of 4863 LUTs. Since the proposed complex divider is fully pipelined and can produce one division result per cycle, the corresponding throughput is 113.3 MOPS. The computation latency is 19 cycles (each CORDIC unit has a latency of 6 cycles plus one cycle delay for

DOI: 10.1587/elex.10.20120879
Received November 20, 2012
Accepted January 21, 2013
Published February 19, 2013

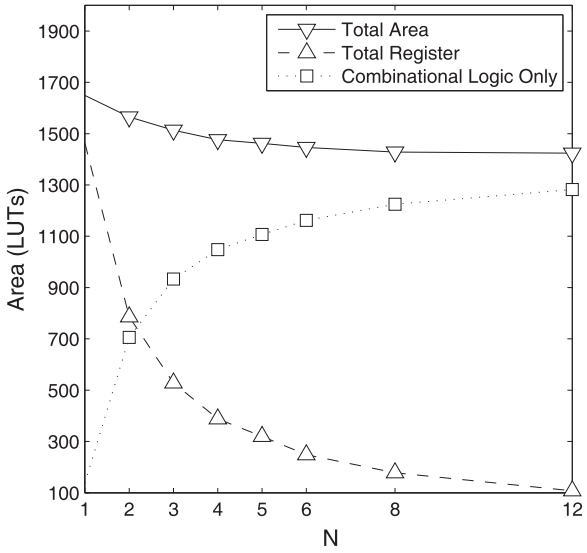

Fig. 3. Synthesized area for CORDIC-I.

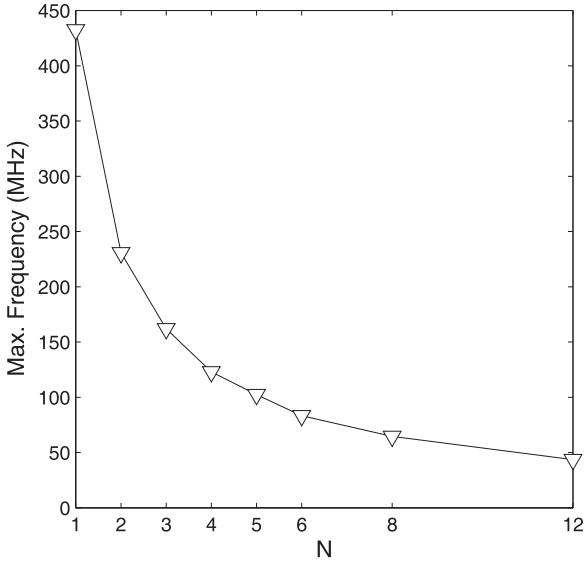

Fig. 4. Reported performance for CORDIC-I.

Table II. FPGA implementation results compared with reference designs.

Scheme	Precision (bit)	Area (LUT)	ROM (Kbit)	Freq. (MHz)	Latency (ns)	Throughput (MOPS)
Proposed	16	4863	-	113.3	167	113.3
Ref. [1]	8	117(slice)	-	100	80	28.6
Ref. [3]	15	527(slice)	-	100	620	1.6
Ref. [4]	16	1185	32	175.9	108	9.3
Ref. [5]	16	1535	96	95.4	94	10.6

input registers of CORDIC-I), which is 167 ns. Compared with scheme [5], which has the highest throughput for 16-bit computation among the references, the throughput of the proposed architecture is improved by more

DOI: 10.1587/elex.10.20120879
Received November 20, 2012
Accepted January 21, 2013
Published February 19, 2013

than 9×. The price is a 2× larger requirement on logic resource. However, the proposed design saves 96 Kbits memories which can further reduce the implemented area when migration to ASICs is considered.

6 Conclusion

In this article, a high-throughput fixed-pointed divider circuit for complex numbers is proposed. By implementing the transformed division formula in Polar coordinates with four optimized CORDIC units, the complex divider can produce one faithfully rounded result per cycle. The 16-bit implementation on Altera Stratix-II FPGA shows a 9× advantage on throughput over the best reference design with only 2× growth on logic cost and saves 96 Kbits memories. Therefore, the proposed complex divider is an ideal choice for DSP applications in which high computation throughput rate and accuracy are often simultaneously required. Besides the presented design, one could also use the area and performance data provided in Section 4 to make quick evaluations for designs of higher throughput or lower costs before actual implementations. Moreover, it should be noted that the proposed structure and design space exploration approach could also be extended to implement fixed-point complex divider of higher precisions.

Acknowledgments

This work is supported by the National Natural Science Foundation of China Grant No. 61106022.

DOI: 10.1587/elex.10.20120879
Received November 20, 2012
Accepted January 21, 2013
Published February 19, 2013