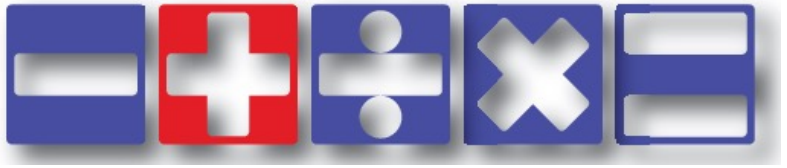
ARITH 2024



HGH-CORDIC: A High-Radix Generalized Hyperbolic Coordinate Rotation Digital Computer

Hui Chen[1], Lianghua Quan[2], Weiqiang Liu[1]

[1] College of Integrated Circuits, Nanjing University of Aeronautics and Astronautics, China

[2] School of Electronic Science and Engineering, Nanjing University, China

31st IEEE International Symposium on Computer Arithmetic
Málaga, Spain. June 10-12, 2024

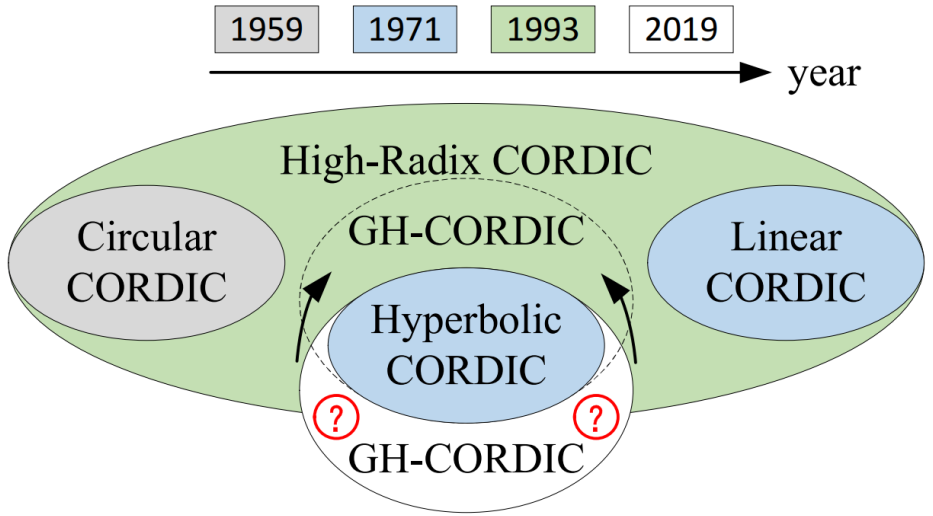
Agenda

- Introduction
- Contribution
- Overview of Radix-2 GH-CORDIC Algorithm
- Definition of High-Radix GH-CORDIC Algorithm
- Software Simulation
- Analysis of Hardware Implementation
- Conclusion

Introduction

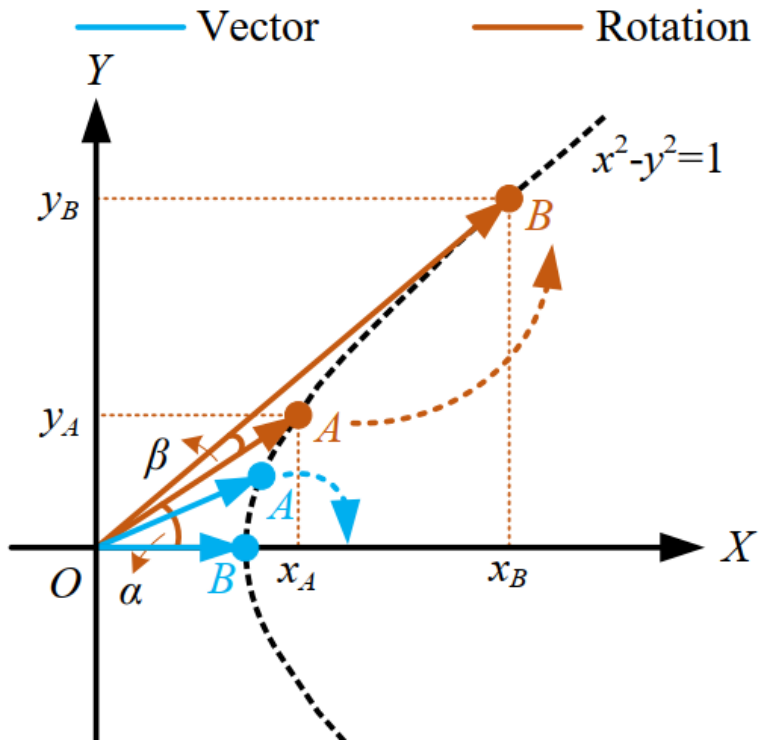
- **Application of CORDIC**
  - (initial) real-time navigation computers for aircraft
  - digital signal processing
  - communication system
  - artificial intelligence
- **Capability of CORDIC**
  - trigonometric, logarithmic, exponential, multiplication, division, and so on
- **Advantage of CORDIC**
  - high-precision, low-complexity

Introduction / Contribution



- **Development and Challenge**
  - Circular CORDIC was first invented by Volder in 1959.
  - Walther developed a unified CORDIC algorithm for the three coordinate systems in 1971.
  - J.D. Bruguera developed an any-radix CORDIC algorithm in three coordinate systems in 1993.
  - Luo *et al*. proposed a generalized hyperbolic CORDIC (GH-CORDIC) algorithm in 2019.
  - CORDIC can achieve higher accuracy through more iterations, but it results in long latency.
- **Research purpose and contribution**
  - To fill the theoretical gap——propose a high-radix GH-CORDIC, which not only can compute the logarithmic and exponential functions with any-base, but also can reduce the number of iterations.

Overview of Radix-2 GH-CORDIC Algorithm



Coordinate rotation principle of radix-2 GH-CORDIC

$$\begin{bmatrix} x_A \\ y_A \end{bmatrix} = \begin{bmatrix} \cosh_b \alpha \\ \sinh_b \alpha \end{bmatrix} = \begin{bmatrix} \frac{b^\alpha + b^{-\alpha}}{2} \\ \frac{b^\alpha - b^{-\alpha}}{2} \end{bmatrix} \quad \text{where } b \in \mathbb{N}^* \text{ but } b \neq 1$$

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} \cosh_b(\alpha+\beta) \\ \sinh_b(\alpha+\beta) \end{bmatrix}$$

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} x_A \cosh_b \beta + y_A \sinh_b \beta \\ y_A \cosh_b \beta + x_A \sinh_b \beta \end{bmatrix}$$

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{1-\tanh_b^2 \beta}}(x_A + y_A \tanh_b \beta) \\ \frac{1}{\sqrt{1-\tanh_b^2 \beta}}(y_A + x_A \tanh_b \beta) \end{bmatrix}$$

$$\beta_i = d_i \tanh_b^{-1}(2^{-i}),\ d_i \in \{-1, +1\},\ i = 1, 2, 3, \cdots$$

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \frac{1}{\sqrt{1-2^{-2i}}} \begin{bmatrix} 1 & d_i 2^{-i} \\ d_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \end{bmatrix}$$

$$\begin{aligned} x_{i+1} &= x_i + d_i(2^{-i} y_i), \\ y_{i+1} &= y_i + d_i(2^{-i} x_i), \\ z_{i+1} &= z_i - d_i \tanh_b^{-1}(2^{-i}), \end{aligned} \qquad d_i = \begin{cases} \text{sign}(z_i), & \text{(for GHR-CORDIC)} \\ -\text{sign}(y_i). & \text{(for GHV-CORDIC)} \end{cases}$$

Overview of Radix-2 GH-CORDIC Algorithm

The modes and outputs of GH−CORDIC

Type	Outputs
GHR-CORDIC	$x_n = K_{gh}(x_0 \cdot \cosh_b(z_0) - y_0 \cdot \sinh_b(z_0))$
	$y_n = K_{gh}(y_0 \cdot \cosh_b(z_0) + x_0 \cdot \sinh_b(z_0))$
	$z_n \to 0$
GHV-CORDIC	$x_n = K_{gh}\sqrt{(x_0)^2 - (y_0)^2}$
	$y_n \to 0$
	$z_n = z_0 + \tanh_b^{-1}(y_0/x_0)$

K_{gh} is the scaling factor, which is defined as:

$$K_{gh} = \prod_{i=1}^{\infty}\left(\sqrt{1-2^{-2i}}\right)$$

Approach of Using GH-CORDIC

$x_0 = 1/K_{gh}$

$y_0 = 1/K_{gh}$

$z_0 = Q$

➡ $y_n = cosh_b(Q) + sinhb(Q) = b^Q$

$x_0 = Q+1$

$y_0 = Q-1$

$z_0 = 0$

➡ $z_n = tanh_b\left(\frac{Q-1}{Q+1}\right) = tanh\left(\frac{Q-1}{Q+1}\right) / ln\, b = \frac{1}{2} log_b\, Q$

Definition of High-Radix GH-CORDIC Algorithm

First, we propose the radix-r GH-CORDIC equations of rotation mode (HGHR-CORDIC)

$$x_{i+1} = x_i + d_i(r^{-i}y_i),$$
$$y_{i+1} = y_i + d_i(r^{-i}x_i),$$
$$w_{i+1} = r(w_i - r^i \tanh_b^{-1}(d_i r^{-i})).$$

Rotation Mode

In order to have the non-zero digits always in the most significant positions to select d_i, here we use a scaled recurrence $w_i = r^i z_i$ to decompose the rotation angle instead of the more conventional recurrence $z_{i+1} = z_i - tanh_b{}^{-1}(d_i r^{-i})$, which is a standard practice in other digit recurrences, such as division and square root [23].

d_i is determined by a selection criterion, which assures the convergence of the proposed HGHR-CORDIC algorithm. We need to bound the new variable w_i by upper limit ($U_i[q]$) and lower limit ($L_i[q]$), which follows a similar method to the one proposed for the radix -r SRT division algorithm [24]. The bounded limits can be defined as follows:

$$U_i[q] = r^i[\tanh_b^{-1}(qr^{-i}) + \frac{p}{r-1}\tanh_b^{-1}(r^{-i})],$$
$$L_i[q] = r^i[\tanh_b^{-1}(qr^{-i}) - \frac{p}{r-1}\tanh_b^{-1}(r^{-i})],$$

where $q \in \{\pm\frac{r}{2}, \pm(\frac{r}{2}-1), \cdots, 0\}$

[23] T. Lang and P. Montuschi, "Very-high radix combined division and square root with prescaling and selection by rounding," in *Proceedings of the 12th Symposium on Computer Arithmetic (ARITH)*, 1995, pp. 124-131.

[24] M. Anane, H. Bessalah, M. Issad, N. Anane, and H. Salhi, "Higher radix and redundancy factor for floating point SRT division," IEEE Trans. VLSI Syst., vol. 16, no. 6, pp. 774-779, 2008.

Definition of High-Radix GH-CORDIC Algorithm

Second, for the radix-r GH-CORDIC equations of vector mode (HGHV-CORDIC), we similarly define a new scaled recurrence $w_i = r^i y_i$. The new iteration equations are

$$x_{i+1} = x_i - d_i(r^{-2i} w_i),$$
$$w_{i+1} = r(w_i - d_i x_i),$$
$$z_{i+1} = z_i + \tanh_b^{-1}(d_i r^{-i}).$$

Vector Mode

To make sure the algorithm is still convergent, w_i also must be bounded between the lower limit function and the upper limit function. Similarly, we define them as

$$U_i[a] = (a + \frac{p}{r-1})x_i, \quad L_i[a] = (a - \frac{p}{r-1})x_i.$$

$d_i = a$ is selected by the criteria given in an interval $L_i[a] \le w_i \le U_i[a]$ to guarantee convergence of HGHV-CORDIC. Similarly, p is often selected as $r/2$.

Based on the new iteration equations, we can get the same convergence result as the radix-2 GH-CORDIC. But most importantly, HGH-CORDIC can quickly calculate the exponential and logarithmic results.

Definition of High-Radix GH-CORDIC Algorithm

Convergence Range of HGH-CORDIC

According to the theorem of CORDIC, the convergence range of HGH-CORDIC depends on the maximum sum of the rotation angles β_{max}, which can be defined as

$$\beta_{max} = \sum_{i=1}^{\infty} |\beta_i| = \sum_{i=1}^{\infty} |\tanh_b^{-1}(d_i r^{-i})|$$
$$= \sum_{i=1}^{\infty} \frac{\tanh^{-1}(d_i r^{-i})}{|\ln b|} = \frac{1}{|\ln b|} \sum_{i=1}^{\infty} \tanh^{-1}(d_i r^{-i}).$$

It is worth mentioning that the high-radix GH-CORDIC should not require any repeated iteration for convergence. Therefore, the number of iterations will be further reduced. Because the maximum value of d_i is $\frac{r}{2}$, the convergence range of HGHR-CORDIC can be given by

$$|z_0| \leq \beta_{max} \leq \frac{1}{|\ln b|} \sum_{i=1}^{\infty} \tanh^{-1}(\frac{r^{1-i}}{2})$$

Similarly, we can get the convergence range of HGHV-CORDIC as follows:

$$|\tanh^{-1}(\frac{y_0}{x_0})| \leq \sum_{i=1}^{\infty} \tanh^{-1}(d_i r^{-i}) \leq \sum_{i=1}^{\infty} \tanh^{-1}(\frac{r^{1-i}}{2}).$$

Definition of High-Radix GH-CORDIC Algorithm

Convergence Range of HGH-CORDIC

When we use HGHV-CORDIC to compute $log_b\, Q$, the input Q meets the following condition.

$$|\frac{Q-1}{Q+1}| \leq \tanh(\sum_{i=1}^{\infty} \tanh^{-1}(\frac{r^{1-i}}{2})).$$

For example, when the radix of HGH-CORDIC is 4 and the base of the generalized hyperbolic function is 2. We can get that

$$|\frac{Q-1}{Q+1}| \leq \tanh(\sum_{i=1}^{\infty} \tanh^{-1}(\frac{4^{1-i}}{2})) = 0.6148.$$

So the range of Q using HGHV-CORDIC can be derived as

$$Q \in [0.2386,\ 4.1924].$$

If we use HGHR-CORDIC to compute b^Q, the input range of z_0 will be

$$|z_0| = |Q| \leq \frac{1}{|\ln 2|} \sum_{i=1}^{\infty} \tanh^{-1}(\frac{4^{1-i}}{2}) = 1.0339.$$

Definition of High-Radix GH-CORDIC Algorithm

Selection Criteria in HGH-CORDIC

The selection function of d_i is crucial, which determines whether HGH-CORDIC can converge. In addition, it also determines the cost of hardware implementation.

Let's start with HGHV-CORDIC. For the sake of derivation and exposition, we take $r = 4$ and $b = 2$ as an example (denoted as H4G2HV-CORDIC). So we get a clear equation to guarantee the convergence of H4G2HV-CORDIC.

$$U_i[a] = (a + \frac{2}{3})x_i,\ L_j[a] = (a - \frac{2}{3})x_i.$$

Then, we can use it to derive the criteria intervals to select d_i. According to the overlap principle, the relationship among selection value, judgment criteria, and overlap interval is as follows.

Selection Value	Judgment Criteria	Overlapping Interval
2	$w_i \geq \frac{3}{2}x_i$	$[\frac{4}{3}x_i,\ \frac{5}{3}x_i]$
1	$\frac{1}{2}x_i \leq w_i < \frac{3}{2}x_i$	$[\frac{1}{3}x_i,\ \frac{2}{3}x_i]$
0	$-\frac{1}{2}x_i \leq w_i < \frac{1}{2}x_i$	$[-\frac{2}{3}x_i,\ -\frac{1}{3}x_i]$
-1	$-\frac{3}{2}x_i \leq w_i < -\frac{1}{2}x_i$	$[-\frac{5}{3}x_i,\ -\frac{4}{3}x_i]$
-2	$w_i < -\frac{3}{2}x_i$	/

Definition of High-Radix GH-CORDIC Algorithm

Selection Criteria in HGH-CORDIC

Similar to the high-radix CORDIC of the circular system, we can also seek an iteration i such that the judgment boundaries belong to the common overlap region. According to the theoretical proof in [25], we can directly get the following inequality:

$$\begin{aligned} L_\infty[2] \le\ & D_i(2) \le U_i[1] && (i \ge 2), \\ L_\infty[1] \le\ & D_i(1) \le U_i[0] && (i \ge 1), \\ L_\infty[0] \le\ & D_i(-1) \le U_i[-1] && (i \ge 1), \\ L_\infty[-1] \le\ & D_i(-2) \le U_i[-2] && (i \ge 2), \end{aligned}$$

where D_i is the selected judgment boundary from the overlap interval. We only need to calculate the judgment boundaries in the first iteration for $D_i[\pm 1]$ and first two iterations for $D_i[\pm 2]$. From this iteration on, the calculated values $D_2(2), D_1(1), D_1(-1), D_2(-2)$ are valid for the remaining iterations.

$$D_i(\pm 1) = \pm\frac{1}{2} \cdot x_1 \quad (\text{if } i \ge 1),$$

$$D_i(\pm 2) = \begin{cases} \pm\frac{3}{2} \cdot x_1 & (\text{if } i = 1), \\ \pm\frac{3}{2} \cdot x_2 & (\text{if } i \ge 2). \end{cases}$$

[25] J. Villalba, E. Zapata, E. Antelo, and J. Bruguera, "Radix-4 vectoring CORDIC algorithm and architectures," J. VLSI Signal Process. Syst., vol. 19, pp. 127-147, 1998.

Definition of High-Radix GH-CORDIC Algorithm

Selection Criteria in HGH-CORDIC

Next, we discuss the selection criteria for H4G2HR-CORDIC ($r = 4$ and $b = 2$). Other cases are similar. To be able to obtain a selection function independent of the iteration index, the limits of the d_i selection intervals must be the same in each iteration. In fact, it is possible for all microrotations with $i \geq 1$.

After deduction and proof, we find that it always has a common overlap region:

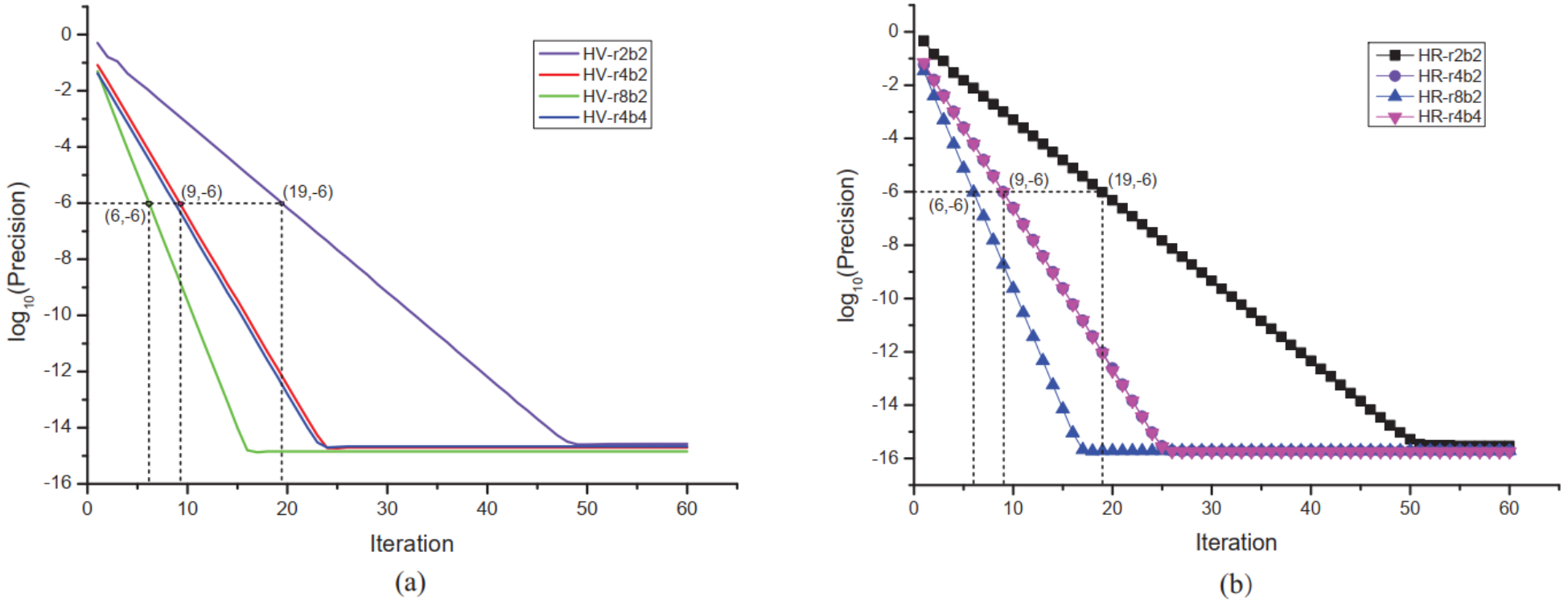
$$2.1873 = L_1[2] \leq D_i(2) \leq U_\infty[1] = 2.4045,$$
$$0.4913 = L_1[1] \leq D_i(1) \leq U_\infty[0] = 0.9618,$$
$$-0.9618 = L_\infty[0] \leq D_i(-1) \leq U_1[-1] = -0.4913,$$
$$-2.4045 = L_\infty[-1] \leq D_i(-2) \leq U_i[-2] = -2.1873,$$

Therefore, for H4G2HR-CORDIC, the relationship among selection value, judgment criteria, and overlap interval are presented as follows.

Selection Value	Judgement Criteria	Overlapping Interval
2	$w_i \geq 2.25$	[2.1873, 2.4045]
1	$0.5 \leq w_i < 2.25$	[0.4913, 0.9618]
0	$-0.5 \leq w_i < 0.5$	[−0.9618, −0.4913]
-1	$-2.25 \leq w_i < -0.5$	[−2.4045, −2.1873]
-2	$w_i < -2.25$	/

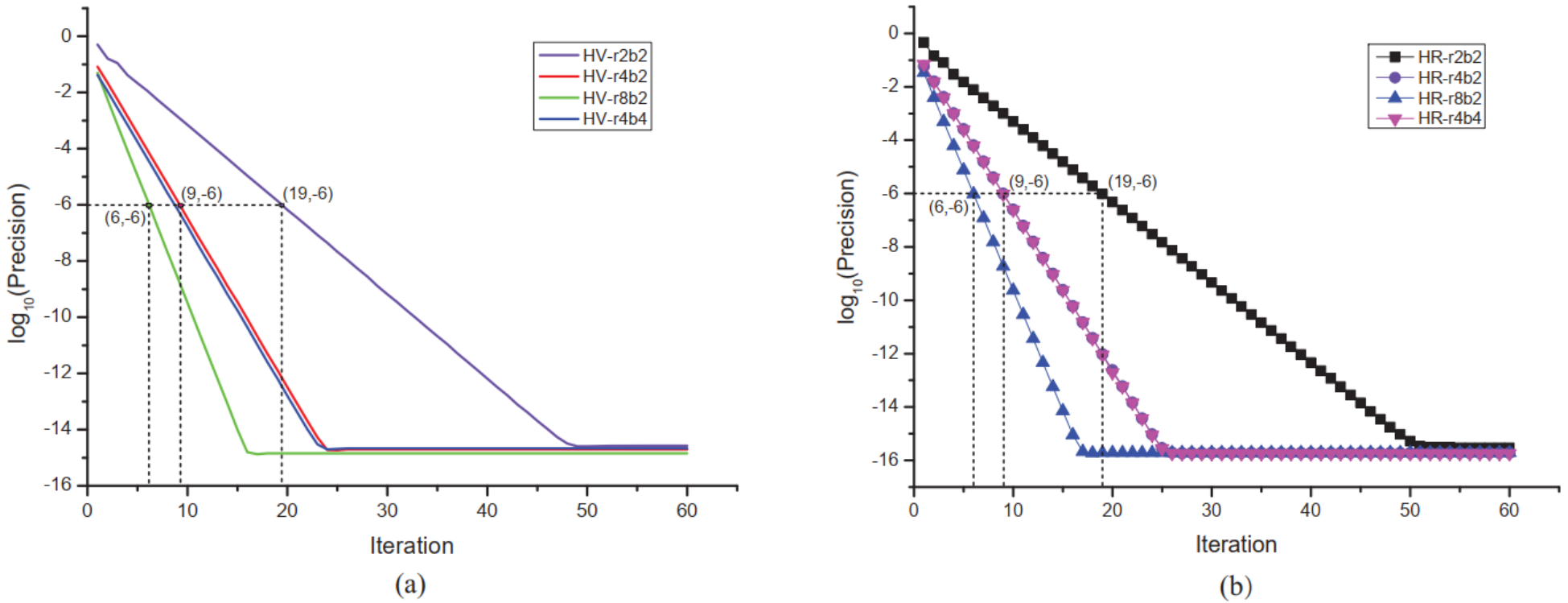
Software Simulation of HGH-CORDIC

To explain the differences between different radix r and whether there are differences between different base b, we choose four cases for simulation using MATLAB and analyze them by control variable method.
The combinations of (r, b) corresponding to these cases are (2, 2), (4, 2), (8, 2), and (4, 4).



Computing $log_b\ x$ and b^x using HGH-CORDIC and traditional GH-CORDIC. (a) Vector Mode; (b) Rotation Mode.

Software Simulation of HGH-CORDIC



Computing $log_b\ x$ and b^x using HGH-CORDIC and traditional GH-CORDIC. (a) Vector Mode; (b) Rotation Mode.

- For each case, we randomly selected 10,000 data from different convergence ranges for testing.
- The results show that the iterations of radix-4 is half of that of radix-2, and that of radix-8 is two-thirds of that of radix-4, which is basically consistent with the results of theoretical derivation.
- In addition, under the condition of radix-4, no matter how *b* changes, its convergence speed and calculation precision are basically same, which is also in line with expectations.

Analysis of Hardware Implementation

Taking the specific case *ARCH*(4, 2) as an example, we will analyze how to efficiently implement the key components, including inverse scaling factor K_{hgh}^{-1}, selection criteria d_i, and inverse hyperbolic tangent function $tanh_2^{-1}$.

Other architectures *ARCH*(r, b) can learn from this idea and analyze them according to the specific situations.

Analysis of the Inverse Scaling Factor

Obviously, calculating K_{hgh}^{-1} is a complicated because d_i is uncertain (d_i^2 has three different values) after each iteration. If it is implemented directly in hardware, it will be very resource-consuming. One of the most common solutions is to take Taylor series expansion and combine it with a smaller LUT.

$$K_{hgh}^{-1} = (1 - d_i^2 4^{-2i})^{-\frac{1}{2}} \approx 1 + \frac{1}{2} d_i^2 4^{-2i} + \frac{3}{8} d_i^4 4^{-4i} + \cdots$$

- For $i > n/4$, K_{hgh}^{-1} can be approximated as 1 because the result of the second term $\frac{1}{2} d_i^2 4^{-2i}$ or more terms is often so small relative to the expected precision that it can be ignored.
- For $i \geq \lfloor n/8 + 1 \rfloor$, K_{hgh}^{-1} can be approximated by the first two terms in n-bit precision.
- For $i \geq \lfloor n/12 + 1 \rfloor$, K_{hgh}^{-1} can be approximated by the first three terms.
- For the remaining iterations, we can implement it with a small LUT.

Analysis of Hardware Implementation

Analysis of the Selection Criteria

- For H4G2HR-CORDIC, we can select multiple combined boundary constant values within the overlap interval.
- But for the sake of simplified hardware implementation, we can choose a uniform set of hardware-friendly compa rison points $\{\pm 0.5,\ \pm 2.25\}$. They require at most 1 sign bit, 2 integer bits, and 2 fractional bits.
- So the width of the comparator only needs to be up to 5 bits.

- For H4G2HV-CORDIC, the choice of d_i is related to the variable x_i.
- From [25], we can reduce the bit width of the comparator without making errors.
- The distance between $U_a(x_i)$ and $D_i(a+1)$ must be greater than or equal to $\frac{1}{6}x_i$ and it must be greater than 2^{-f}, where f refers to the truncated fractional bits.
- Assume that the input range of $log_2 x$ is [1, 2], that is, the initial value range of x_0 of H4G2HV-CORDIC is [2, 3]. Through analysis, the minimum value of x_i in all iterations is 2 and the maximum value is 3 ($max\{w_i\}$=5.43). We can deduce that $f > 1.585$ and we need at least 2 fractional bits.
- Therefore, we totally assimilate 6 bits for x_i and w_i, including 2 fractional bits, 3 integer bits, and 1 sign bit.

[25] J. Villalba, E. Zapata, E. Antelo, and J. Bruguera, "Radix-4 vectoring CORDIC algorithm and architectures," J. VLSI Signal Process. Syst., vol. 19, pp. 127-147, 1998.

Analysis of Hardware Implementation

Analysis of the Inverse Hyperbolic Tangent Function

- No matter which mode HGH-CORDIC works in, they must calculate $tanh_b^{-1}(d_i r^{-i})$. If we calculate it directly, it is very complicated.
- The general method is to use LUT. As the number of iterations increases, the size of LUT increases exponentially. To reduce the LUT size, we can use an approximate scheme by Taylor series expansion of $tanh_b^{-1}x$.

$$\tanh_b^{-1} x = \frac{1}{lnb}(x + \frac{1}{3}x^3 + \frac{1}{5}x^5 + \cdots)$$

- When $i \geq \lceil n/6 \rceil$, $tanh_b^{-1}(d_i r^{-i})$ can be approximated to $d_i r^{-i} / ln\ b$. In this case, w_{i+1} and z_{i+1} will

$$w_{i+1} = r(w_i - \frac{1}{lnb} d_i),\ z_{i+1} = z_i + \frac{1}{lnb}(d_i r^{-i})$$

  where $\frac{1}{ln\ b}$ is a concrete constant. When $d_i \neq 0$, there are only two possible results for $\frac{r|d_i|}{ln\ b}$ or $\frac{|d_i|}{ln\ b}$.
- When i is less than $\lceil n/6 \rceil$, we can calculate the complicated term $r^{i+1} tanh_b^{-1}(d_i r^{-i})$ or the term $tanh_b^{-1}(d_i r^{-i})$ in advance, and then select them by looking up the table.
- It is worth learning that we can also use the zero-skipping technique [25] (used in high-radix circular CORDIC) to further reduce the number of iterations (about 20%) and resources.

[25] J. Villalba, E. Zapata, E. Antelo, and J. Bruguera, "Radix-4 vectoring CORDIC algorithm and architectures," J. VLSI Signal Process. Syst., vol. 19, pp. 127-147, 1998.

Analysis of Hardware Implementation

Analysis of Performance

Item	[17]	[27]	Proposed
Function	$log_b x$, b^x	lnx, e^x	$log_b x$, b^x
Radix	2	$r \geq 8$	$r \geq 4$
Iteration	$n + m$	$\lceil \frac{n - log_2 R}{log_2 r} \rceil + 1$	$\lceil \frac{n + log_2 (r/2)}{log_2 r} \rceil$

m: The total repeated iterations, when $i = 4, 13, 30, \cdots$
R: R needs to be less than r.

- From the previous work [14], we can observe that when designing a pipelined hardware architecture, the area of radix-4 CORDIC will be smaller than that of radix-2 due to the decreasing number of cascades.
- Radix-8, on the other hand, has a small gain because the implementation complexity increases dramatically but the number of iterations decreases disproportionately.
- Compared with [27], we can support the logarithmic and exponential operations with arbitrary base, and also support radix-4, which has the most cost-effective performance and hardware overhead.
- Compared with the traditional GH-CORDIC [17], we can reduce more iterations to speed up the calculation of arbitrary $log_b\, x$ and b^x.

[14] J. Rudagi and S. Subbaraman, "Comparative analysis of radix-2, radix-4, radix-8 CORDIC processors," in Int. Conf. Inventive Comput. Informat. (ICICI), 2017, pp. 378-382.
[17] Y. Y. Luo *et al.*, "Generalized hyperbolic CORDIC and its logarithmic and exponential computation with arbitrary fixed base," IEEE Trans. VLSI Syst., vol. 27, no. 9, pp. 2156-2169, 2019.
[27] E. Antelo, T. Lang, and J. D. Bruguera, "Very-high radix CORDIC rotation based on selection by rounding," J. VLSI Signal Process. Syst., vol. 25, pp. 141-153, 2000.

Conclusion

- Compared with applied research, the development speed of theoretical research is relatively lagging behind. To fill the gap in theoretical research, we propose a theory of high-radix generalized hyperbolic CORDIC for more applications.
- Because base-2 logarithmic and exponential functions are the most commonly used nonlinear functions in floating-point conversions, and radix-4 CORDIC is the best cost-effective algorithm in hardware, we focus on the specific case (r=4, b=2) to demonstrate the theory of HGH-CORDIC.
- Then we elaborate on the feasibility and practicability of the theory through simulation and analysis.
- Finally, the advantages of HGH-CORDIC are summarized through comparative analysis, such as low latency and high precision, which are usually difficult to achieve at the same time.
- In the future, we will conduct research on its hardware implementation based on this theory, especially to explore the low-latency and high-precision hardware architectures for its related applications.

Thank you very much for your attention!