Dual Fixed-Point CORDIC Processor: Architecture and FPGA Implementation

Andres Jacoby and Daniel Llamocca
Electrical and Computer Engineering Department
Oakland University
Rochester, MI, USA
ajacobykratell@oakland.edu, llamocca@oakland.edu

Abstract
We introduce Dual Fixed Point CORDIC, that provides a compromise between Fixed Point and Floating Point CORDIC hardware implementations. A fully parameterized hardware is presented that allows for extensive exploration of the resources-accuracy design space, from which we generate optimal (in the multi-objective sense) realizations. We compare Fixed Point, Dual Fixed Point, and Floating Point CORDIC units in terms of resources and accuracy. Results show the effectiveness of Dual Fixed Point for CORDIC implementation where the increase in resources is largely offset by the high accuracy improvements.

Key Contributions
- Parameterized architecture validated on a FPGA
- Design Space Exploration
- Comparisons among DFX, FX, and FP architectures

Methodology and Architectures
We used the expanded CORDIC algorithms to implement the DFX Hyperbolic and Circular CORDICs. The expanded hyperbolic CORDIC is described mathematically by:

\[ \begin{align*}
    x_i &= x_{i-1} - b_i y_{i-1} (1 - 2^{-i}) \\
    y_i &= y_{i-1} - b_i x_{i-1} (1 - 2^{-i}) \\
    z_i &= z_{i-1} - b_i z_{i-1} (1 - 2^{-i})
\end{align*} \]

where \( b_i \) is the \( i \)-th bit of the input number, \( x_{i-1} \), \( y_{i-1} \), and \( z_{i-1} \) are the state variables, and \( z_i \) is the actual result of the \( i \)-th iteration.

An \( n \)-bit Dual Fixed-Point (DFX) number is composed of a \( (n-1) \)-bit signed significand \( x \) and an exponent bit \( E \). The exponent determines the scaling for the significand:

\[ x = \begin{cases} 
\text{num} \cdot 2^{E-1} & \text{if } E = 0 \\
\text{num} \cdot 2^{E+1} & \text{if } E = 1
\end{cases} \]

DFX adder/subtractor architecture includes a pre-scaler, an FX adder/subtractor, and a post-scaler. The pre-scaler aligns the DFX input operands so they can be added in FX arithmetic. In the post-scaler, the range detector determines whether the result is a \( \text{num}0 \) or \( \text{num}1 \); we then select the proper result and set the exponent bit.

The function \( x^y = e^{y \ln x} \) is computed in two steps:
1. We first get \( x_2 = (\ln x) / 2 \) followed by \( x_2 \times 2y = y \ln x \).
2. Then, we use \( x_{i+1} = y_{i+1} \) and \( x_{i+1} = x_{i+1} \mod e \) to get \( x_{i+1} = e^{y \ln x} = x^y \).

The argument bounds of \( x^y \) for which \( x^y \) converges are given by \( |y \ln x| \leq \gamma \) of (8).

Results
The Pareto-optimal realizations for \( \ln x \) allows us to only consider optimal hardware realizations while simultaneously satisfying accuracy and resource constraints. For \( \ln x \), if we want highest accuracy and fewer than 1k resources, we would pick DFX48 [43, 29]. Table 3 describes how DFX compares to FX in terms of resources and accuracy. For \( x^y \) on average, resources increased by 55% while accuracy improved 61.45dB. Fig. 6 shows the relative error of DFX and two FX realizations each with the same \(\phi_0 \) or \(\phi_1 \). Table 4 lists resources an accuracy values of FP and DFX units for \( e^x, x^y \). The resource increase and accuracy improvement of the FP units over the DFX units. For \( e^x \), a 53% resource increase yields a 108.48dB gain in accuracy.

Conclusion
We presented and validated fully customized DFX CORDIC and \( \ln x, x, x^y \) units. We extensively explored the accuracy-resources design space and extracted the Pareto front. Comparisons between DFX, FX, and FP CORDIC architectures demonstrate that DFX is a good candidate for Dual Fixed Point CORDIC implementation: DFX accuracy improvements more than make up for the resource increase with respect to FX. Further work will focus on the implementation of scale-free CORDIC that requires fewer iterations for the same range of convergence, and by leveraging Partial Reconfiguration technology to implement Dynamic Dual Fixed Point for even larger dynamic ranges.