



Electrical and Computer Engineering EE3623 Embedded Systems Design Laboratory



Implementation of Square Root with the CORDIC LogiCORE Block

Dennis Silage, PhD

The Coordinate Coordinate Rotation Digital Computer (CORDIC) is a simple and efficient algorithm to calculate hyperbolic and trigonometric functions, vector rotations and translation and the square root. The CORDIC is commonly used when no hardware multiplier is available as the only operations it requires are addition, subtraction, bit shifting and table lookup. However, when a hardware multiplier is available, power series calculations for these functions are generally faster than the CORDIC algorithm.

The modern CORDIC algorithm was first described in 1959 by Jack E. Volder but it is similar to techniques published by Henry Briggs as early as 1624. John Stephen Walther at Hewlett-Packard further generalized the algorithm.

The CORDIC Graphical User Interface (GUI) contains four screens for configuration:

1. Screen 1: Used to configure the functional selection and architecture of the CORDIC algorithm LogiCORE.
2. Screen 2: Used to configure the phase and magnitude data formats, optional control signals, and synchronization.
3. Screen 3: Provides options for configuring the rounding mode and input-outputs.
4. Screen 4: Provides options for advanced configuration parameters: coarse rotation, iterations, internal precision, compensation scaling and displays the latency and controls the inclusion of LogiCORE instantiation.

The task for this two week Laboratory for Thursday April 24 and May 1st are as follows:

1. The initial task of this Laboratory is to implement the CORDIC algorithm logiCORE block for the square root of an unsigned positive integer in the range $0 \leq X_{IN} < 2^{16}$ or 16-bit data. The data input is set in the 8 slide switches. BTN0 indicates valid data and BTN1 indicates the most significant 8 bits are inputted. BTN0 without BTN1 indicates that the least significant 8 bits are inputted. The data input is to be displayed on the first line of the LCD and the square root results on the second line.

- The second task of this Laboratory is to cascade two CORDIC algorithm LogiCORE to calculate the 4th root (square root of a square root) from the data inputted as in Task 1. Data transfer between CORDIC modules is coordinated by the RDY (ready) signal from the first module. The data input is to be displayed on the first line of the LCD and the square root results on the second line.

For both of these tasks describe the apparent *resolution* and *round-off error* of the square and 4th root observed for a representative range of unsigned 16-bit integer numbers. You need to research the definition of the terms *resolution* and *round-off error*. Can this be controlled by the choice of the CORDIC algorithm LogiCORE implementation? If so, how and what are the results obtained?

Much of these tasks can be performed in simulation before hardware synthesis and verification for the slides switches, buttons and LCD are implemented and would support you Laboratory report. The design reuse aspect of existing modules for these peripherals simplifies the tasks.

The CORDIC algorithm is described in the Xilinx data sheet DS249 and posted on Blackboard. This Laboratory thus consists of an exercise in the specifications of a LogiCORE block and the implementation of the specific LogiCORE block as a nested module in a Xilinx ISE project.

