ELEC 522 Project 4 Report

Matthew Nutt

October 30, 2025

My implementation of the CORDIC algorithm (`cordic.cpp`) attempts to minimize duplicate hardware by parameterizing the rotation direction metric in the main loop. That is, rotations in CORDIC are the same whether in the Sin/Cos mode or the Arctan mode, but the difference between the two is that the angle is used to determine the rotation direction in Sin/Cos mode, whereas the y value is used in Arctan mode. My HLS design shares the logic for rotations while preserving this difference in heuristic.

This architectural design does help to minimize duplicate hardware, but it comes at the cost of overhead in the CORDIC loop.

There wasn't much room to apply optimization pragmas, but I did include a pipelining pragma for the CORDIC loop as well as an array partitioning pragma for the angle lookup table. These helped cut latency in half.

Timing and resource utilization post-synthesis for `cordic.cpp`.

My Vitis testbench verifies the functionality of the design:

```
========== Sin/Cos Tests Begin ==========
Test: Sin/Cos of 0 radians
HW Results:
X: 1 Y: 0 (Angle: 0.00012207)
SW Results:
X: 1 Y: 0
Expected Results:
X: 1 Y: 0
----------
Test: Sin/Cos of -pi/4 radians
HW Results:
X: 0.706909 Y: -0.707275 (Angle: 0.00012207)
SW Results:
X: 0.707107 Y: -0.707107
Expected Results:
X: 0.70711 Y: -0.70711
----------
Test: Sin/Cos of pi/2 radians
HW Results:
X: 0.00012207 Y: 0.999878 (Angle: 0.00012207)
SW Results:
X: 4.89659e-12 Y: 1
Expected Results:
X: 0 Y: 1
----------
Test: Sin/Cos of -pi radians
HW Results:
X: -1.00037 Y: 0 (Angle: 0.00012207)
SW Results:
X: -1 Y: -2.06823e-13
Expected Results:
X: -1 Y: 0
----------
Test: Rotation of (2, 1) by pi radians
HW Results:
X: -2.00012 Y: -0.999878 (Angle: 0.00012207)
SW Results:
X: -2 Y: -1
Expected Results:
X: -2 Y: -1
```

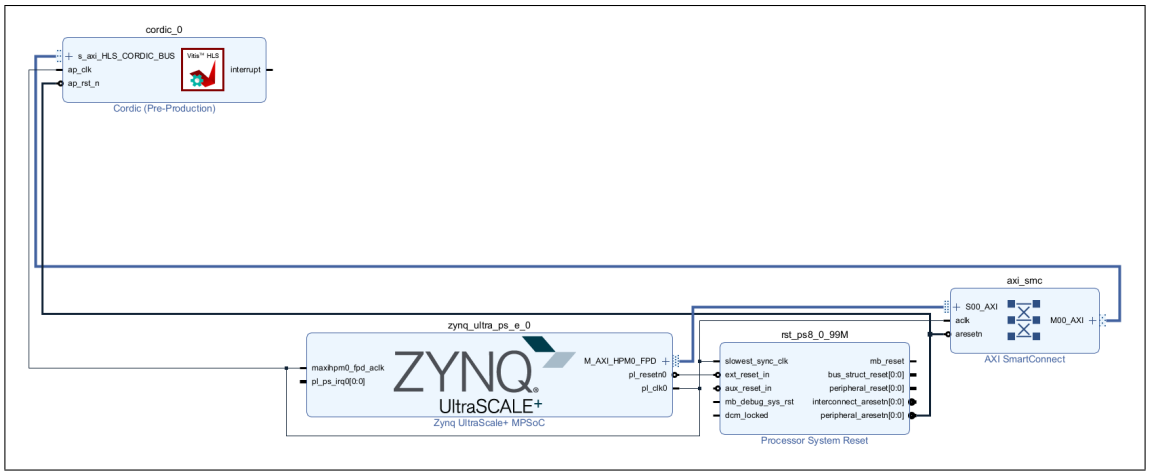
Successful testbench results for Sin/Cos mode.

```
========== Arctan Tests Begin ==========
Test: Arctan of (1/sqrt 2, 1/sqrt 2)
HW Results:
X: 1.00024 (Y: 0) Angle: 0.785522
SW Results:
X: 1 Angle: 0.785398
Expected Results:
X: 1 Angle: 0.78540
----------
Test: Arctan of (1, -1)
HW Results:
X: 1.41431 (Y: 0) Angle: -0.785278
SW Results:
X: 1.41421 Angle: -0.785398
Expected Results:
X: 1.41421 Angle: -0.78540
----------
Test: Arctan of (0, 1)
HW Results:
X: 1 (Y: 0) Angle: 1.57092
SW Results:
X: 1 Angle: 1.5708
Expected Results:
X: 1 Angle: 1.57080
----------
Test: Arctan of (-1/sqrt 2, -1/sqrt 2)
HW Results:
X: 1.00024 (Y: 0) Angle: 0.785522
SW Results:
X: 1 Angle: 0.785398
Expected Results:
X: 1 Angle: 0.78540
```

Successful testbench results for Arctan mode.

These results aren't exactly identical to the native C++ functions, but they are often accurate within 3 decimal places. With 3 integer bits and 13 decimal bits, the theoretical limit of the fixed point datatype I used is 4 decimal places.

After the design was finalized in Vitis, it was packaged for synthesis and implementation in Vivado. Various screenshots from this process can be seen below.



Vivado block diagram of the system.

Resource	Utilization	Available	Utilization %
LUT	1589	70560	2.25
LUTRAM	2	28800	0.01
FF	1088	141120	0.77

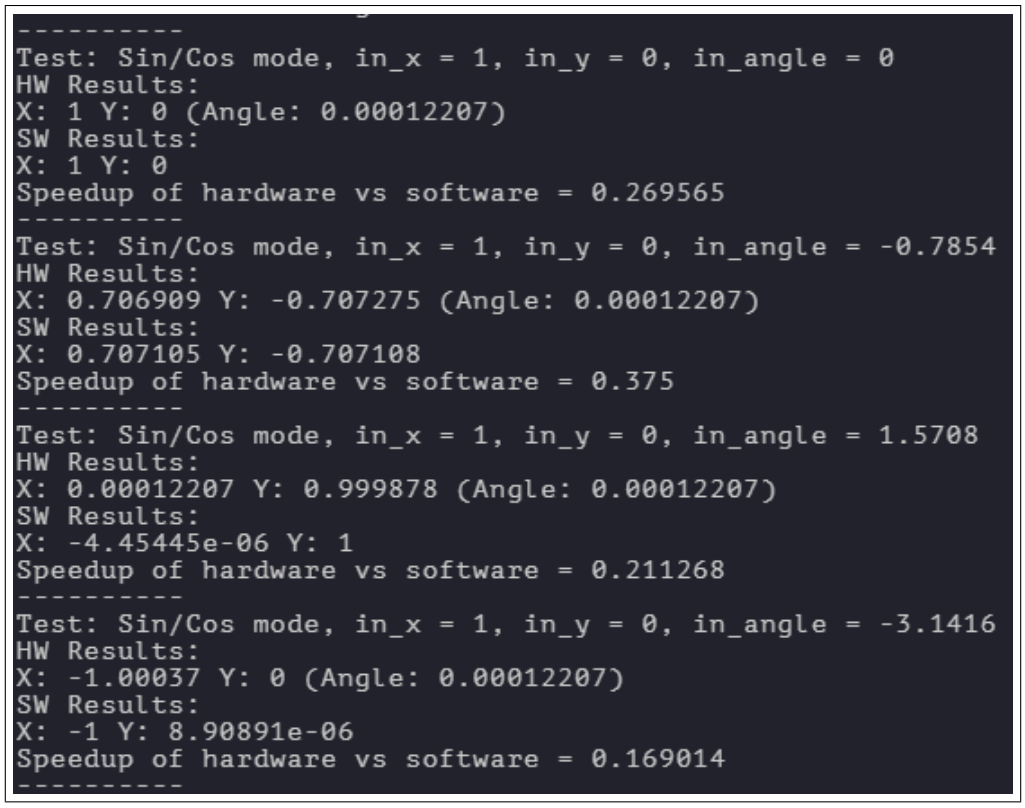
Vivado resource utilization post-implementation.

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.253 ns	Worst Hold Slack (WHS):	0.014 ns	Worst Pulse Width Slack (WPWS):	3.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	2604	Total Number of Endpoints:	2604	Total Number of Endpoints:	1091

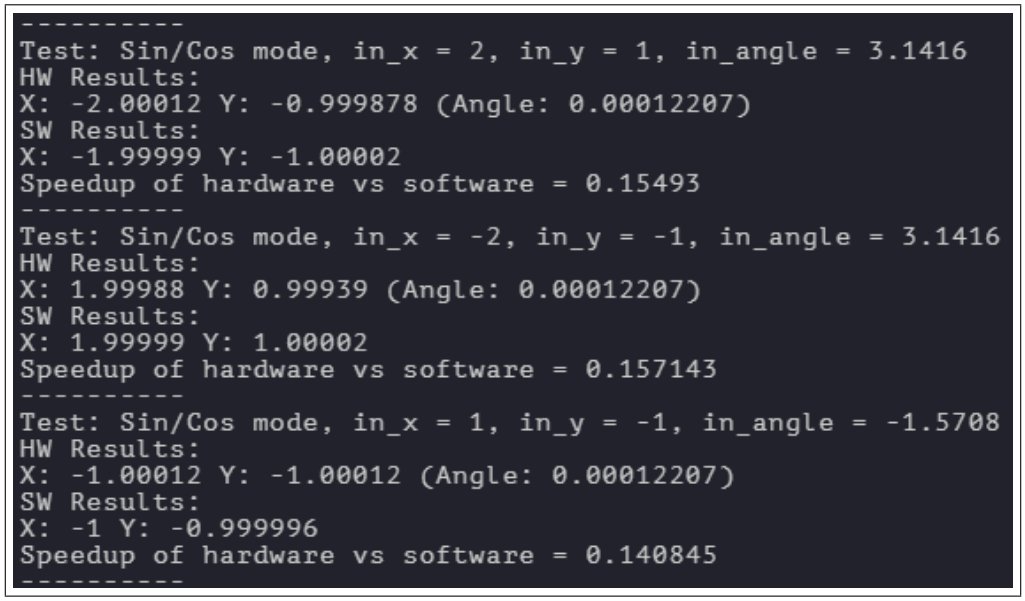
All user specified timing constraints are met.

Vivado timing performance post-implementation.

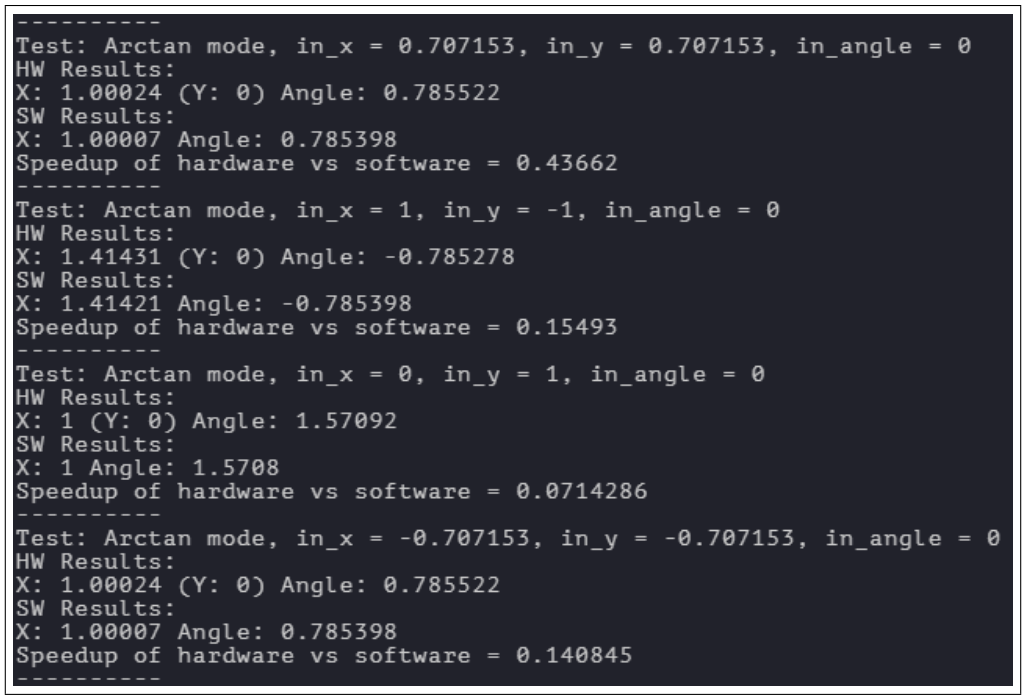
From the Vivado implementation, the hardware platform was ported over to Vitis. An application, `main.cc`, was created for testing the design once implemented on a AUP-ZU3 board. This application included data reinterpretation functionality for handling the fixed-point datatype over the AXI interface. The results of this testing, viewed via serial monitor, can be seen below.



Serial monitor output for Sin/Cos tests.



Serial monitor output for rotation tests.



Serial monitor output for Arctan tests.