Laboratory 5

(Due date: **005**: March 31st, **006**: April 1st, **007**/**008**: April 2nd)

OBJECTIVES

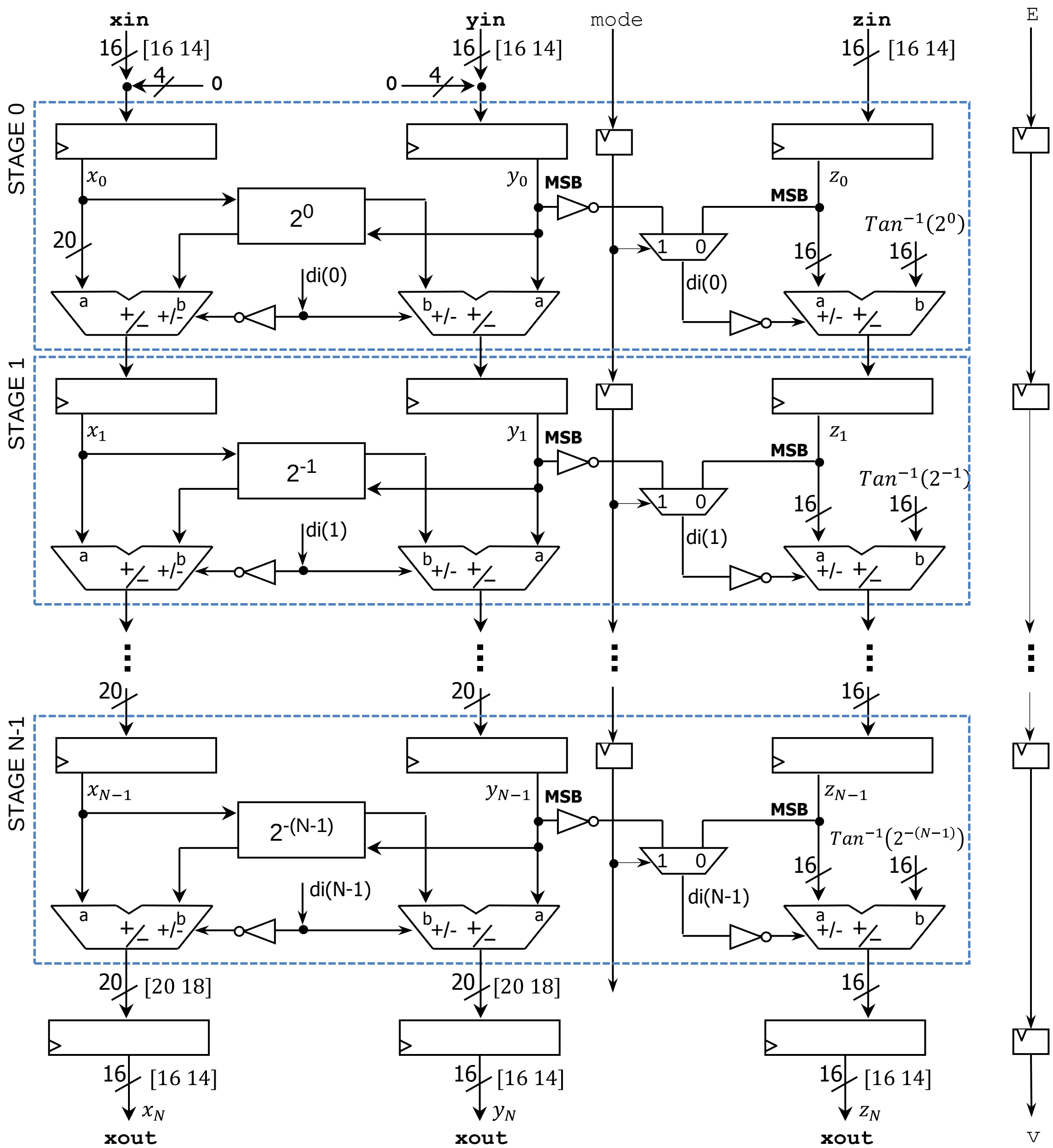
- ✓ Design a pipelined circuitry for trigonometric circuit (CORDIC) in Fixed-point Arithmetic.
- ✓ Test the CORDIC circuit using real data represented in Fixed-Point Arithmetic.
- ✓ Learn how to read input and output text files for Simulation.

VHDL CODING

- ✓ Refer to the Tutorial: VHDL for FPGAs for parametric code for: adder/subtractor and register.

FIRST ACTIVITY: PIPELINED FX CIRCULAR CORDIC (60/100)

- Design the pipelined Circular CORDIC architecture (basic range of convergence) with N iterations ($i = 0,1,2,\dots,N-1$) shown in the figure below. $mode$ = '0' → Rotation Mode. $mode$ = '1' → Vectoring Mode.
- The circuit must be written in parametric VHDL code with N as the only parameter. $N = 4$ *to* 16.
- Tip: Implement a stage i as a parametric component. Then on the top file, just instantiate N of those components.
- **Angles**: They are represented in the format [16 14]. Units: radians.

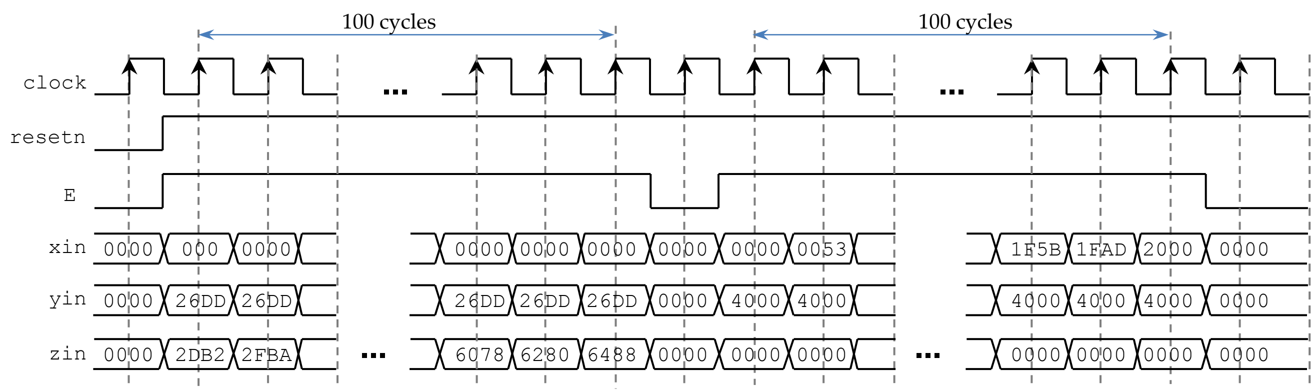


Instructor: Daniel Llamocca
TAs: Andrew DeJonge, Nathan Kelley

SECOND ACTIVITY: SIMULATION (40/100)

- Create a testbench that reads input values (x_0, y_0, z_0) from input text files and writes output values (x_N, y_N, z_N) on output text files. You must test the parameterized CORDIC circuit for $N = 12$ and $N = 16$. The testbench must:
  - ✓ Read input values $(x_0,\ y_0,\ z_0)$ from two input text files (provided):
    - `in_benchR.txt`: Data for Rotation Mode testing.
      100 data points $(x_0,\ y_0,\ z_0)$. Data format: [16 14]. Each line per data point written as hexadecimals: `|x0|y0|z0|`.
      Data set: $x_0 = 0, y_0 = 1/A_n, z_0 = -\pi/2\ to\ \pi/2$. z_0: 100 equally-spaced values between $-\pi/2\ to\ \pi/2$.
      With this data set in the rotation mode, note that $x_N \rightarrow -sin(z_0)$, $y_N \rightarrow cos(z_0)$.
    - `in_benchV.txt`: Data for Vectoring Mode testing.
      100 data points $(x_0,\ y_0,\ z_0)$. Data format: [16 14]. Each line per data point written as hexadecimals: `|x0|y0|z0|`.
      Data set: $x_0 = 0.0\ to\ 0.5, y_0 = 1, z_0 = 0$. x_0: 100 equally-spaced values between $0.0\ to\ 0.5$.
      With this data set in the vectoring mode, note that $x_N \rightarrow A_n\sqrt{x_0^2 + y_0^2}, z_N \rightarrow atan(y_0/x_0)$.
  - ✓ Write output results $(x_N,\ y_N,\ z_N)$ on `out_bench_N12.txt` and `out_bench_N16.txt`. Data format: [16 14], each line per data point written as hexadecimals: `|xN|yN|zN|`. Each output text file should have 200 data points (100 from the rotation mode and 100 from the vectoring mode). Using a handful of data points, verify that your results are correct.

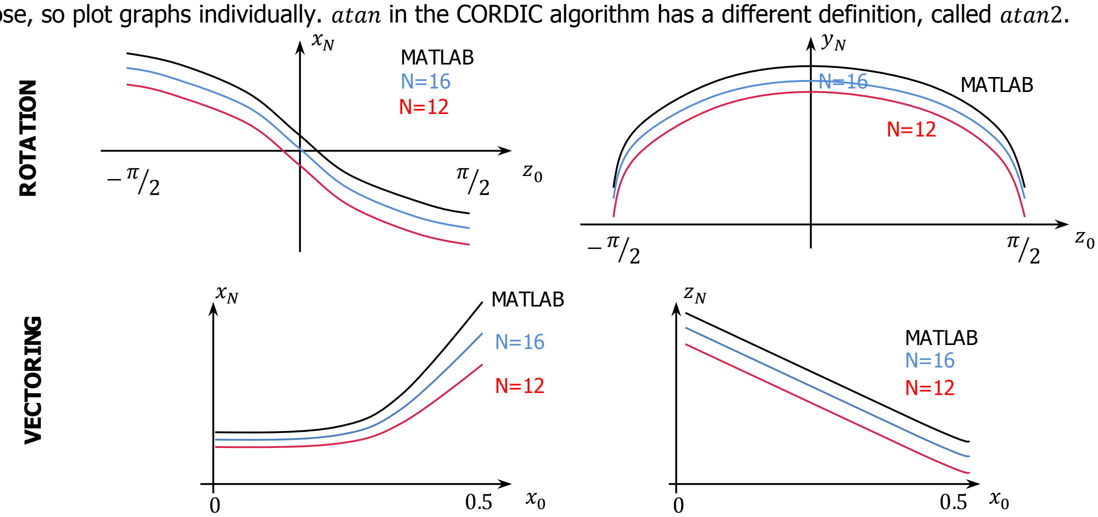
- Your testbench must fed data at the rate of one set per cycle (as this is a pipelined circuit). See figure below for reference. Output data should appear at the same rate (after N+1 cycles since the first data was entered).



- Tips:
  - ✓ Vivado: Make sure that the input text files are loaded as simulation sources.
  - ✓ Vivado: The output text file should appear in `sim/sim_1/behav`.
  - ✓ Vivado: To represent data as fixed-point numbers, use Radix → Real Settings in the Vivado simulator window.
  - ✓ For reference, the following MATLAB script can be useful: cordic_example_ece4710.m. It generates the input text files and reads the output textfile (out_bench.txt) as specified here. It uses the Circular CORDIC MATLAB/Octave model.

OPTIONAL THIRD ACTIVITY: PROCESS DATA READ FROM OUTPUT TEXT FILES (+10)

- Read data from output text files in MATLAB® (or Octave). Convert data to their corresponding real numbers.
- Plot the results (rotation and vectoring modes for $N = 12{,}16$) along with the function values (Rotation: $x_N \rightarrow -sin(z_0)$, $y_N \rightarrow cos(z_0)$. Vectoring: $x_N \rightarrow A_n\sqrt{x_0^2 + y_0^2}, z_N \rightarrow atan(y_0/x_0)$ to which the CORDIC results should converge. The values will be very close, so plot graphs individually. $atan$ in the CORDIC algorithm has a different definition, called $atan2$.



- Compute the MSE for each case. This metric assesses how close the hardware results are to the ideal ones (function values to which the CORDIC results converge). The smaller the MSE, the more accurate the hardware results are.

	Rotation		Vectoring	
	N=12	N=16	N=12	N=16
x_N				
y_N				
z_N				

$$MSE = \frac{1}{T}\sum_{i=1}^{T}\left(D_i - \widehat{D}_i\right)^2$$

- Submit (as a .zip file) the following files: VHDL code, VHDL testbench, and output text files to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.

TA signature: ______________________________ **Date:** ___________________________