Observations:

Arbitrary Precision:

Precision	Performance	DSP	Pass/Fail
ap_fixed<24,2>	Total CPU user time: 8.06 seconds. Total CPU system time: 1.45 seconds. Total elapsed time: 19.41 seconds; peak allocated memory: 332.531 MB.	7	Pass
ap_fixed<20,2>	Total CPU user time: 7.45 seconds. Total CPU system time: 1.41 seconds. Total elapsed time: 18.74 seconds; peak allocated memory: 323.832 MB.	7	Pass
ap_fixed<15,2>	Total CPU user time: 7.82 seconds. Total CPU system time: 1.52 seconds. Total elapsed time: 19.26 seconds; peak allocated memory: 325.754 MB	8	Pass
ap_fixed<10,2>			Fail

Question 1:

# of Rotations	Performance	Resource Usage	Accuracy of Results
10	Error @ C Sym	N/A	Theta Off by .001
12	Total CPU user time: 5.97 seconds. Total CPU system time: 1.39 seconds. Latency: 132 cycles	peak allocated memory: 252.477 MB. DSP: 10	Theta off by .003 +/- .001
15	Total CPU user time: 5.96 seconds. Total CPU system time: 1.45 seconds. Latency: 162 cycles	peak allocated memory: 252.328 MB. DSP: 10	Accurate
18	Error @ C Sym		
20	Error @ C Sym		

Why does the accuracy stop improving after some number of iterations?
Not all " Kvalues" and "angle" values are being used in the calculation. The printf statement essentially rounds the result by only printing 4 decimal places for the float value.

Can you precisely state when that occurs?
That occurs at 14 iterations.

Question 2:
Is one data type sufficient for every variable or is it better for each variable to have a different type?

Precision	Performance	DSP	Pass/Fail
ap_fixed<17,2>	Total CPU user time: 7.18 seconds. Total CPU system time: 1.52 seconds. Total elapsed time: 18.59 seconds; peak allocated memory: 322.703 MB. Latency 116 cycles	7	Pass

It would be better for each variable to have a different type, if the input and expected output values are not all the same. While testing data types with different precision, I noticed I would get error with data type for int, since the expected return value is float and precision is important to the calculation.

Does the best data type depend on the input data?
I found that the output data is more important, as we don't want to lose precision by rounding errors.

What is the best technique for the designer to determine the data type(s)?
The designer should determine which data types would be most helpful by considering the calculations that are being done, how to optimize the calculation and what is the expected output.

Question 3:
What is the effect of using simple operations (add and shift) in the CORDIC as opposed to multiply and divide?
The CORDIC calculation is done much faster, as the bitwise shifting to multiply/divide by a power of 2 is executed much faster.

How does the resource usage change? Performance? Accuracy?

Precision	Performance	DSP	Pass/Fail
ap_fixed<32,16>	Total CPU user time: 5.85 seconds. Total CPU system time: 1.12 seconds. Total elapsed time: 16.88 seconds; peak allocated memory: 318.000 MB. Latency: 113 cycles	5	Pass

Question 4: (Referring to LUT Implementation)

- *How does the input data type affect the size of the LUT?*

  The size of the LUT is dependent on word length (in bits) of the fixed-point representation of the input data type.

  How does the output data type affect the size of the LUT? Precisely describe the relationship between input/output data types and the number of bits required for the LUT.

  The LUT is copying over the values of the output data, therefore their data types need to match. LUT_SIZE is then 2^16 for both.

- *The testbench assumes that the inputs x, y are normalized between [-1,1]. What is the minimum number of integer bits required for x and y?*

  Two bits, one for the sign and one for the integer value

- *What is the minimal number of integer bits for the output data type R and Theta?*

  1 integer bit

- *Modify the number of fractional bits for the input and output data types. How does the precision of the input and output data types affect the accuracy (RMSE) results?*

Fractional Bits	RMSE(R)	RMSE(Theta)
5	0.023094084113836	0.051045715808868
3	0.086298309266567	0.170881420373917
7	0.947205603122711	1.751199364662170

- *What is the performance (throughput, latency) of the LUT implementation. How does this change as the input and output data types change?*

Data Type	Latency
ap_fixed<8,3>	20ns
ap_fixed<8,5>	20ns
ap_fixed<8,0>	20ns

- *What advantages/disadvantages of the CORDIC implementation compared to the LUT-based implementation*

  With the CORDIC implementation, you do not need the expected values (for LUT). But the calculation has a larger latency.