Project2: CORDIC

Cheng-Yu Chen

Architectures:

Baseline: It uses loop unrolling and data type of all data is ap_int<27,3>. Also, it uses adds and shifts to multiply and its report shows that there is no DSP48E.

Optimized1: I choose the one with 11 rotation since its RMSE(theta) < 0.001 and it use less resources.

Optimized2: Data type of some data is changed to ap_int<25,1> or ap_int<26,2>.

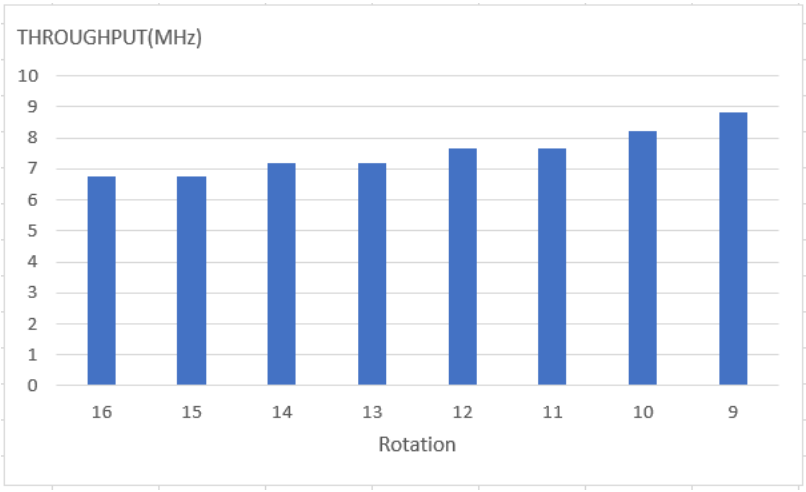
Optimized3: It uses multiplier to multiply.

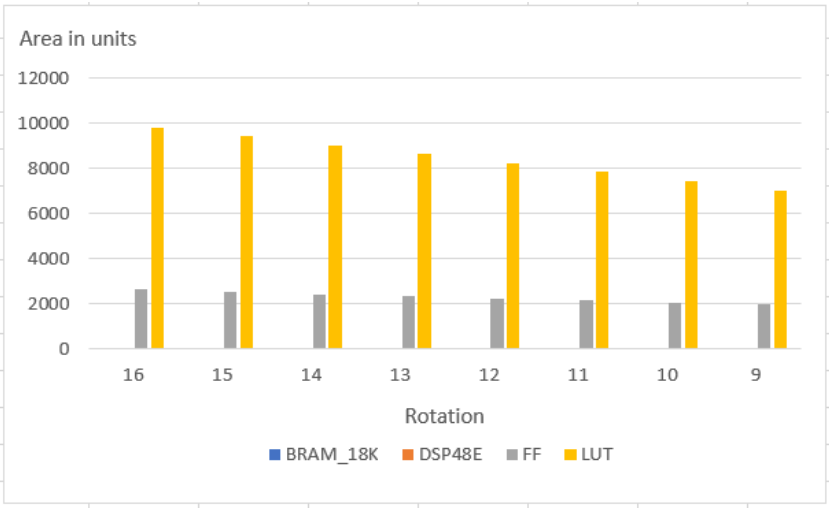
Optimized4: It replace if-else statement with operator ?.

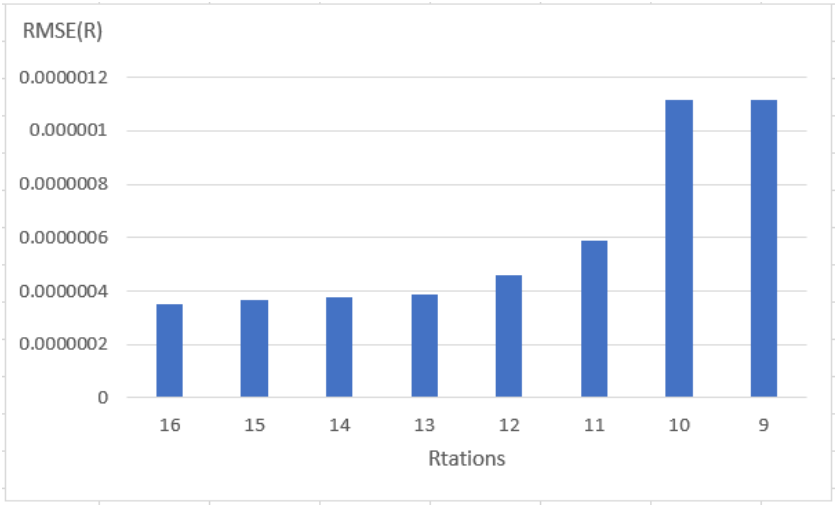
Testbench: I add two test cases in every quadrant so there are 12 test cases totally.

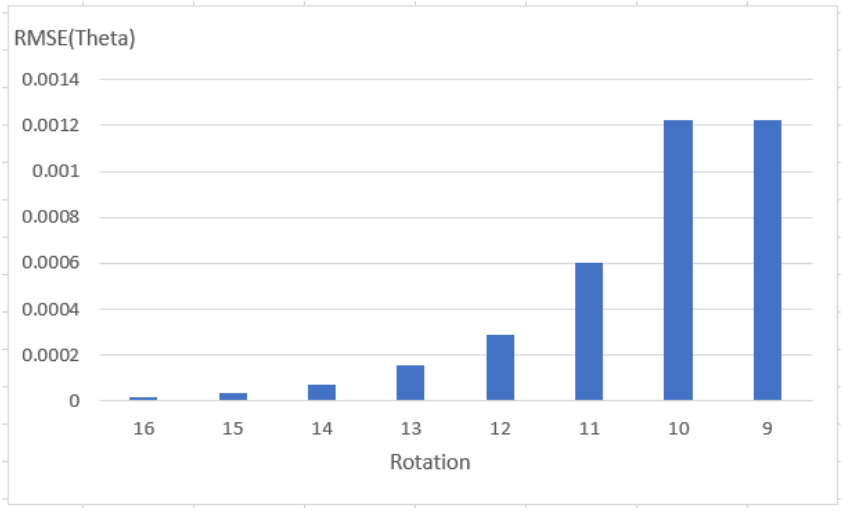
Question 1:

One important design parameter is the number of rotations. Change that number and describe the results. What happens to performance? Resource usage? Accuracy of the results?









Throughput will gradually increase as rotation decreases. On the other hand, all resources will decrease as rotation decreases since the circuit needs less components. However, the accuracy will decrease as rotation decrease.

Why does the accuracy stop improving after so many iterations?

It's because the range of rotation would reduce after every iteration. In other words, the change of theta and r would gradually reduce and they are gradually close to the real value. Therefore, when they are very close to the real value, the accuracy would not change much.

Can you precisely state when that occurs?

Since the data type of output is float and float is 24 bits precision, the accuracy would stop changing when the change amount of theta is under 2^{-24}, which would occur when it's after 24 iterations(when theta is close to zero, arctan(theta)=theta). Or, if stopping improving means RMSE(theta) or RMSE(r) is under 0.001, it occur after 11 iterations.

Question 2:

Another important design parameter is the data type of the variables. Is one data type sufficient for every variable or is it better for each variable to have a different type?

	baseline	optimized2
BRAM_18K	0	0
DSP48E	0	0
FF	2631	2603
LUT	9808	9689
THROUGHPUT(MHz)	6.761325	6.773003

Using only one data type will waste some resources so it's better that each data have its own data type by only considering its own range of data. For example, in this project, some data are always less than 1 so I specify ap_int<25,1> to them.

Does the best data type depend on the input data?

Yes, if we know the range of input data, we could know the range of every data in the function since we know our function. So, knowing the input data could help us to specify the best data type to each data in the function.

What is the best technique for the designer to determine the data type?

The designer should consider the maximum and minimum of every data so they could specify best data type to each data and they could optimize the latency and resources. For example, if we know the range of some data is between 1000 and 0, we could specify them to ap_int<10> since the range of ap_int<10> is between 1023 and 0.

Question 3:

What is the effect of using simple operations (add and shift) in the CORDIC as opposed to multiply and divide?

It could reduce resources of the circuit since multiplier and divider are very complex circuits. Also, it could shorter the latency of the circuit since add and shift are simple.

How does the resource usage change? Performance? Accuracy?

	baseline	optimized3
BRAM_18K	0	0
DSP48E	0	2
FF	2631	2382
LUT	9808	11399
THROUGHPUT(MHz)	6.761325	4.997501
RMSE(R)	3.49474E-07	2.28251E-07
RMSE(Theta)	1.69502E-05	1.69502E-05

Optimized3 would need more some resources such as LUT and DSP48E and have lower throughput. However, it needs less FF and have higher accuracy of R.

Question 4:

How does the ternary operator '?' synthesize?

It is synthesized as multiplexers since multiplexers and '?' have the same function. If the condition is true, they both will assign one value to the output. If not, they both will assign the other value to the output.

Is it useful in this project?

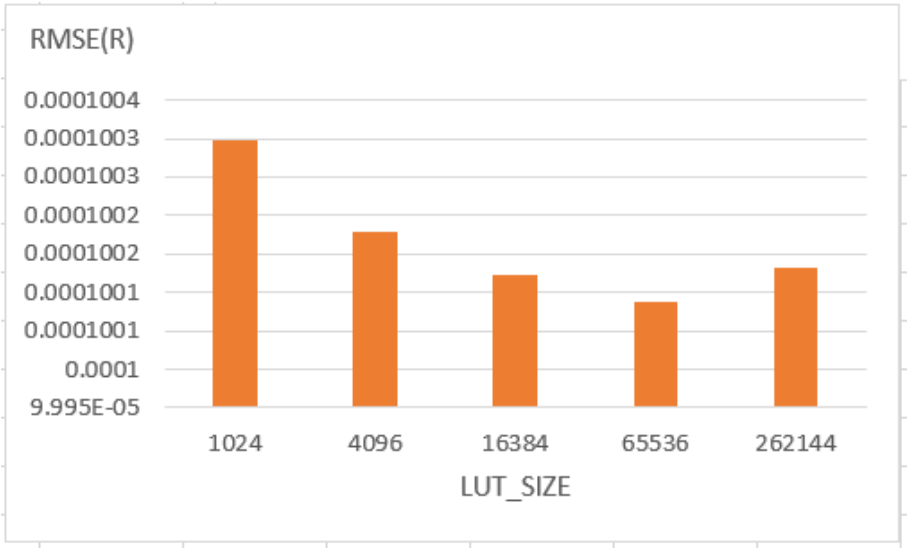
	baseline	optimized4
BRAM_18K	0	0
DSP48E	0	0
FF	2631	2091
LUT	9808	8883
THROUGHPUT(MHz)	6.761325	6.784721
RMSE(R)	3.49474E-07	3.49474E-07
RMSE(Theta)	1.69502E-05	1.69502E-05

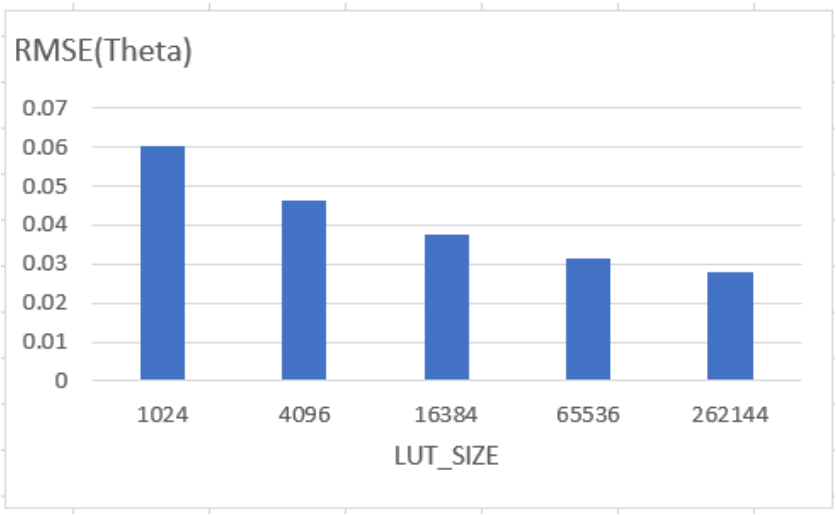
Yes, according to the report, using operator ? could reduce some expression so optimized4 have less FF and LUT, which leads to higher throughput.

Question 5:

These questions all refer to the LUT-based CORDIC: Summarize the design space exploration that you performed as you modified the data types of the input variables and the LUT entries. In particular, what are the trends with regard to accuracy (measured as error)?

LUT_SIZE vs error:



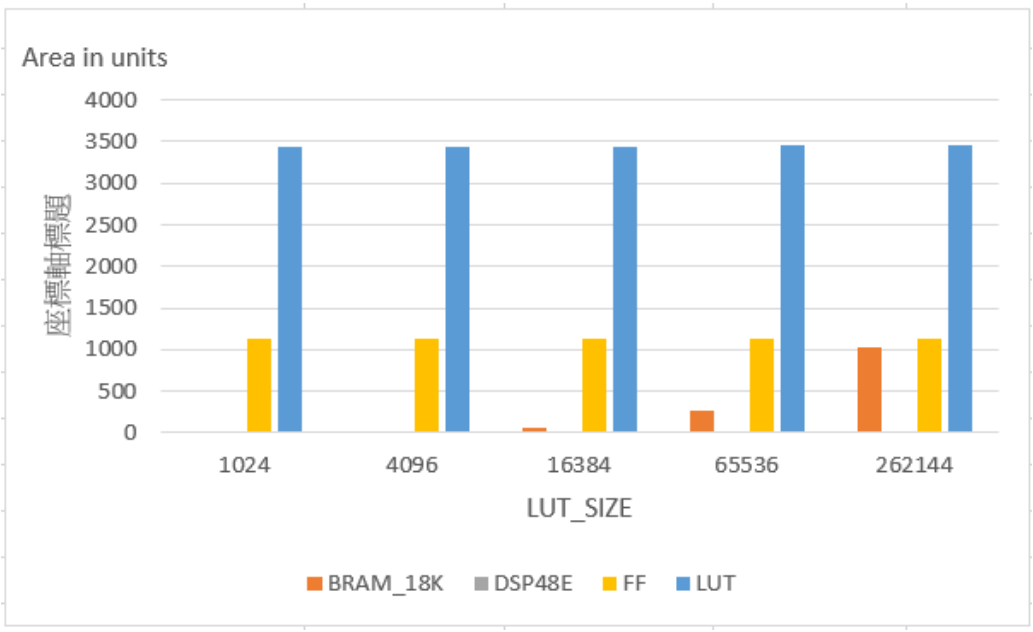


Size of fixed-point vs error: RMSE(R)=0.000100123, RMSE(Theta)=0.037400421 for all size of fixed-point

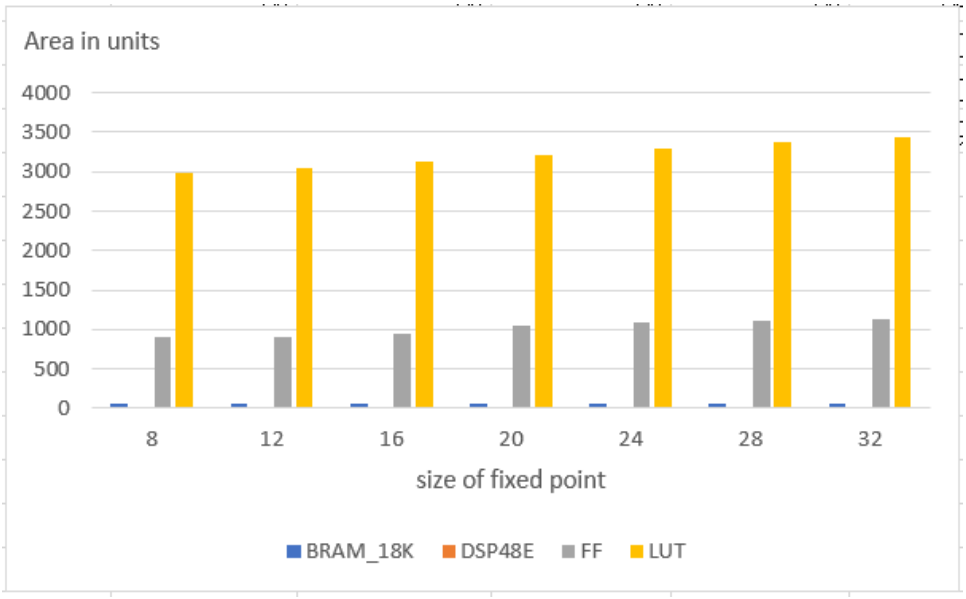
As LUT_SIZE increases, the accuracy of R and Theta increases. However, the accuracy of R and Theta never changes as the size of fixed-point changes.

How about resources?

LUT_SIZE vs area:



Size of fixed-point vs area:

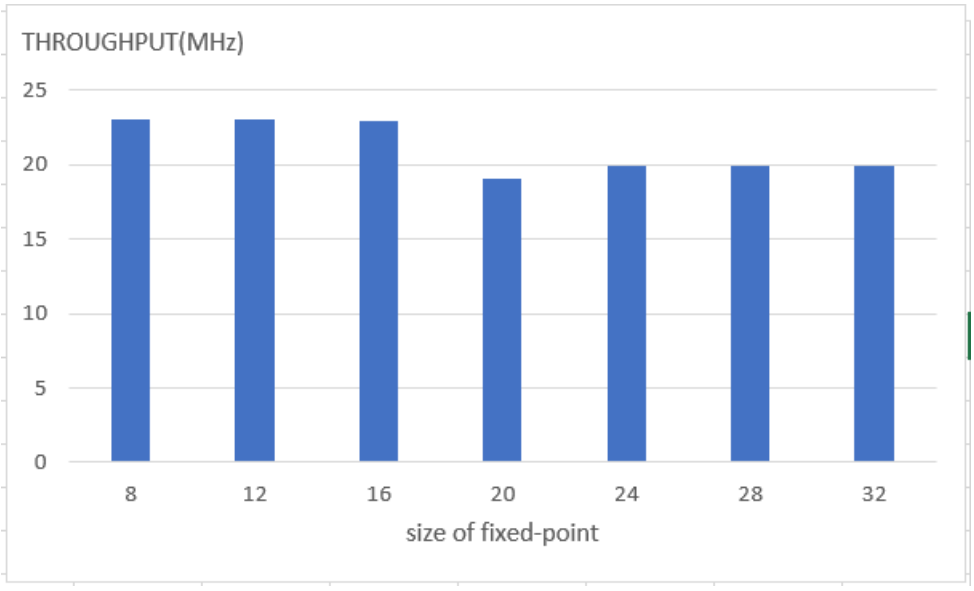


As LUT_SIZE increases, the area of LUT and FF slightly increase and the area of BRAM exponentially increases. On the other hand, as the size of fixed-point increases, the area of LUT and FF slightly increases but the area of BRAM never changes.

What about the performance?

LUT_SIZE vs throughput: 8.377MHz for all LUT_SIZE

Size of fixed-point vs area:



As LUT_SIZE increases, the throughput stays the same. On the other hand, as the size of fixed-point increases, the throughput decreases and at some point, it would decrease sharply since the latency of the circuit change.

Is there a relationship between accuracy, resources, and performance?

As the accuracy increases, the resources have to increase since the circuit would need more LUT but this wouldn't affect the throughput. With the fixed accuracy, throughput will decrease as the resources increase since there will be some unnecessary latency.

What advantages/disadvantages does the regular CORDIC approach have over an LUT-based approach?

Advantage: Regular CORDIC approach need less resources such as LUT since it could just use some expression to calculate the result. Also, it could attain higher accuracy without too much cost since it just takes little more time to calculate the result.

Disadvantage: Regular CORDIC approach have lower throughput since LUT-based approach just only has to look up the table to attain the result, which takes very little time.