University of Tehran
Electrical and Computer Engineering Department
Computer Aided Design – Fall 94
CA4: CORDIC Module Design

In this assignment you are to implement a CORDIC coprocessor to calculate $\tan^{-1}$ of a given input value. According to Figure 1 CORDIC module has 2 control signal, start and reset, 16-bit input data and 16-bit output angle. Complete explanation of CORDIC algorithm can be found in Appendix I.

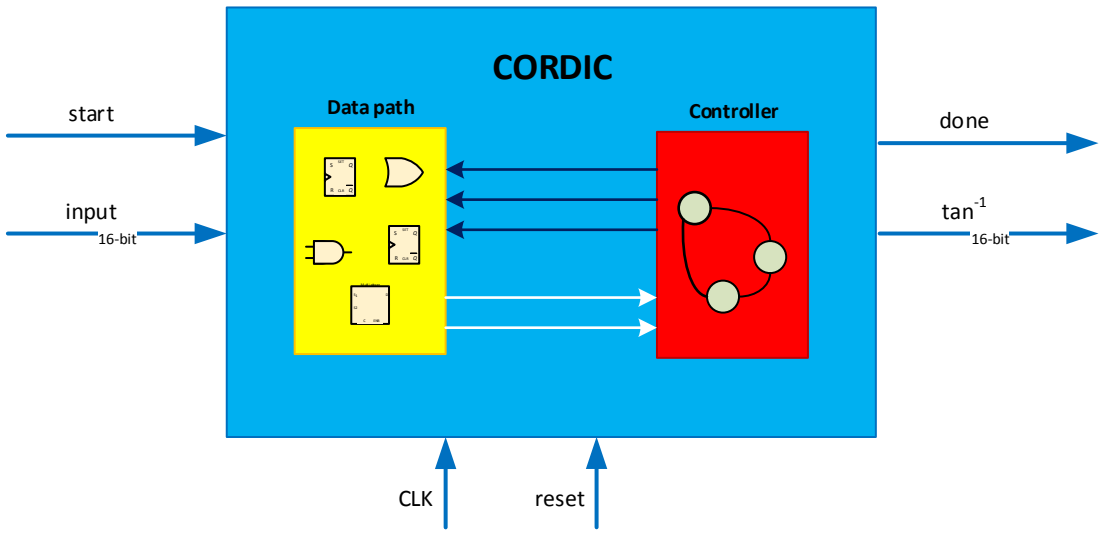

Figure 1: CORDIC Schematic

Figure 1 depicts the block diagram of $\tan^{-1}$ module. The $\tan^{-1}$ module gets a fixed-point value of *"input"* after start signal is asserted. Following this, computation is started. When the result is available on output, *"done"* signal is asserted. At any phase of computation when *"reset"* signal goes high, all of the internal registers are set to default value.

You should assume that input data is a 16-bit signed fixed-point data with 4 bit fraction. Output is also 16-bit signed fixed-point with 13 bit fraction.

Hints:

- You should specify number of cycles in your design according to the bit-width of the system.
- You should handle cases that overflow occurs in add/subtract module. (saturate data when overflow occurs)
- Bit-width of intra system is not necessarily same as system port
- You should use look-up table for step angles.

Notice that:

- You must follow RTL design flow (controller and data path).
- Your design must be parameterized.
- You should handle *"start"*, *"reset"* and *"done"* signals properly.
- In your test bench, consider various scenarios in a single run.
- Your design must be synthesizable.

You have two weeks to submit your assignment. The report must be well-organized and you should include clear schematic of your data path and state machine with necessary and clear explanations.

Appendix I: CORDIC algorithm

CORDIC is an efficient algorithm of implementation for a variety of mathematical functions like $\tan^{-1}$ function. This algorithm uses basic operations e.g. addition, subtraction, comparison, shifting and lookup table.

In the following the use of CORDIC algorithm in vector mode is explained. This type of CORDIC algorithm can be used to calculate $\tan^{-1}\theta$, $\cot^{-1}\theta$ and $\sqrt{a^2+b^2}$. The matrix representation of this rotations is shown below.

$$\begin{bmatrix} x_R \\ y_R \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix}$$

Rotation can be performed in multiple steps.

$$\begin{bmatrix} x_{j+1} \\ y_{j+j} \end{bmatrix} = \begin{bmatrix} \cos\theta_j & -\sin\theta_j \\ \sin\theta_j & \cos\theta_j \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}$$

$$\begin{bmatrix} x_{j+1} \\ y_{j+j} \end{bmatrix} = \cos\theta_j \begin{bmatrix} 1 & -\tan\theta_j \\ \tan\theta_j & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}$$

If we use θ_j so that $tanh\theta_j$ equals a power of two, we can implement the multiplications of $tanh\theta_j$ with bit shift. Thus we use θ_j as:

$$\theta_j = \tan^{-1} 2^{-j}$$

And the relation between steps can be shown as:

$$\theta = \sum \sigma_j \theta_j \, , \sigma_j \in \{-1, +1\}$$

And thus we have:

$$\tan\theta_j = \sigma_j 2^{-j}$$

With these equation we have:

$$\begin{bmatrix} x_{j+1} \\ y_{j+j} \end{bmatrix} = \cos\theta_j \begin{bmatrix} 1 & -\sigma_j 2^{-j} \\ \sigma_j 2^{-j} & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}$$

Since the total transfer function is derived from multiplication of all transfer functions, we can apply the scaling $\cosh\theta_j$ at the final step only, as:

$$K = \prod_{j=0}^{\infty} \left(1 - 2^{-2j}\right)^{1/2} \approx 1.6468$$

Using the equations above, if we apply scaling factor at the final step, we can say that:

$$\begin{cases} x_{j+1} = x_j - \sigma_j 2^{-j} y_j \\ y_{j+1} = y_j + \sigma_j 2^{-j} x_j \end{cases}$$

Now, we need an extra variable to represent the remaining portion of θ that has not been applied yet. We call this variable z_j whose sign determines the value of σ_j as:

$$z_{j+1} = z_j - \sigma_j \theta_j$$

$$\sigma_j = \begin{cases} +1, & y_j < 0 \\ -1, & y_j \geq 0 \end{cases}$$

We continue the steps of the algorithm until $y_j = 0$, then we have:

$$x_f = K(x_{in}^2 + y_{in}^2)^{1/2}$$
$$y_f = 0$$
$$z_f = z_{in} + \tan^{-1}\left(\frac{y_{in}}{x_{in}}\right)$$

For which, at each step we calculate x_j, y_j and z_j as:

$$\begin{cases} x_{j+1} = x_j - \sigma_j 2^{-j} y_j \\ y_{j+1} = y_j + \sigma_j 2^{-j} x_j \\ z_{j+1} = z_j - \sigma_j \theta_j \end{cases}$$

The initial values of algorithm to calculate $\tan^{-1}$ of given input:

$$\begin{cases} x_1 = {}^{1}/_{K} \\ y_1 = {}^{a}/_{K} \\ z_1 = 0 \end{cases}$$

That results in:

$$\begin{cases} x_f = \sqrt{a^2 + 1} \\ y_f = 0 \\ z_f = \tan^{-1} a \end{cases}$$

Here is an example of $\tan^{-1} 1$:

Table 1: Example of CORDIC for exponential function

step	x	y	z	sign	Step angle
0	0.607238	0.607238	0	-1	0.785398
1	1.214477	0	0.785398	-1	0.463648
2	1.214477	-0.60724	1.249046	1	0.244979
3	1.366286	-0.30362	1.004067	1	0.124355
4	1.404239	-0.13283	0.879712	1	0.062419
5	1.412541	-0.04507	0.817293	1	0.03124
6	1.413949	-0.00093	0.786053	1	0.015624
7	1.413963	0.021166	0.77043	-1	0.007812
8	1.414129	0.01012	0.778242	-1	0.003906
9	1.414168	0.004596	0.782148	-1	0.001953
10	1.414177	0.001834	0.784101	-1	0.000977
11	1.414179	0.000453	0.785078	-1	0.000488
12	1.414179	-0.00024	0.785566	1	0.000244
13	1.414179	0.000108	0.785322	-1	0.000122