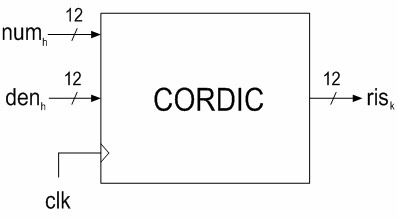
Report on design of a CORDIC device for arctan

# Introduction

The objective of this project is to design a CORDIC (COordinate Rotation DIgital Computer) device to compute arctan.



The relationship to implement is

## The CORDIC algorithm

The CORDIC algorithm provides a mean to compute trigonometric functions iteratively using only shifts and adds, which are cheap to implement in a dedicated electronic device. It has two sets of equation:

* Rotation, where a vector iteratively rotated by a specified angle
* Vectoring, where a vector is iteratively rotated to an angle of 0 rad to measure the original angle.

For the computation of arctangent, only the latter is needed. The equations are as follows:

Where if , otherwise. These equations hold for any decreasing sequence of angles.   
The use of allows to use only right shifts to compute and . To compute a LUT must be used. The LUT must be sized to the number of iterations to execute.

For we have

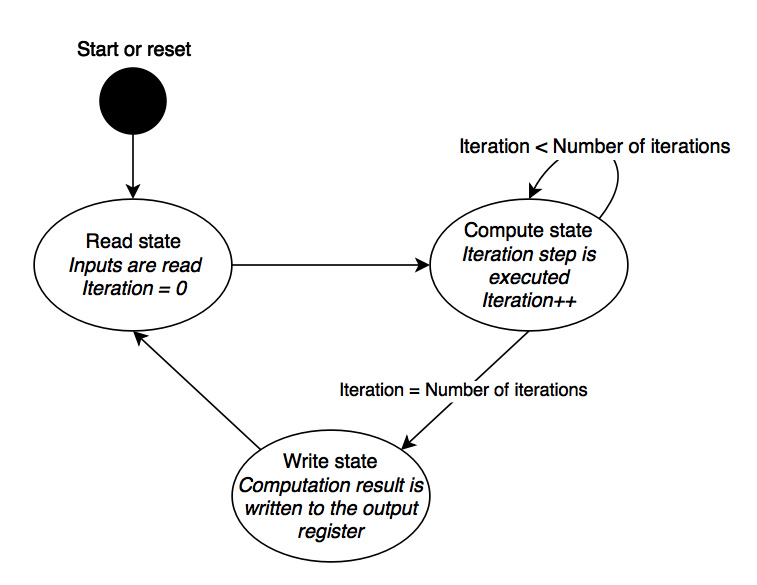
Where is a gain that can be ignored, while is the value we aimed to compute given that .

# Design phase

## Device structure

Since the algorithm to implement is iterative, we designed the device as a finite state machine with three main states:

* Read state, where the inputs are taken to start a new computation and the internal registers are reset
* Compute state, where the iterative steps are executed
* Write state, where the result of the computation is written to the output register



We identified 5 main components that make up the device:

* A core FSM, which manages the register, the status and the mapping of the other components
* Three combinatorial networks, which get and to compute and .
* A LUT that stores the values of at index .

These components were designed as generics, that is with parameters that can be set when instantiated. These parameters specify the bit resolutions of input and output values as well as the number of iterations to execute.

## Design choices

The result of the algorithm is expected to be a real number . To represent these values in the output of the device, we chose the signed fixed point representation. Given as the bit resolution for the output, the first two most significant bits are used to represent the sign and integer part, while the rest is used to represent the fractional part. This way rational numbers can be represented with various degree of precision depending on .

Due to the sums computed in the device there’s a risk of incurring in overflows for particularly high inputs. To avoid posing additional restraints on the device user, we chose to internally extend the input bit resolution by 2.