CORDIC Functional (Trigonometric + Hyperbolic, IEEE-754 Floating Point)

**Inputs**

* clk, reset → Clock & Reset
* zi (32-bit IEEE-754 float) → Input angle (in degrees)
* control\_th → Mode select:
  + 1 → Trigonometric functions (sin, cos, tan …)
  + 0 → Hyperbolic functions (sinh, cosh, tanh …)
* start\_control → Start signal

**Outputs (32-bit IEEE-754 floating point)**

* done → High when operation complete
* sin\_th\_x → sin/sinh result
* cos\_th\_x → cos/cosh result
* tan\_th\_x, cot\_th\_x, sec\_th\_x, cosec\_th\_x → Derived ratios

**Major Blocks**

1. **Precomputed Constants Storage**
   * two\_po\_ni[0:19] → IEEE-754 encoded values of 2⁻ᶦ
   * alpha\_i\_tr[0:19] → IEEE-754 atan(2⁻ᶦ) values (degrees)
   * alpha\_i\_tr\_hi[0:19] → IEEE-754 atanh(2⁻ᶦ) values (radians)
2. **Initialization Unit (Floating Point)**
   * Trig Mode (control\_th=1):
     + xi = 0.60565f (CORDIC scaling constant, IEEE-754 format)
     + zi = input angle (already IEEE-754 degree)
   * Hyperbolic Mode (control\_th=0):
     + xi = 1.0f (IEEE-754 0x3F800000)
     + Convert input degrees → radians (using floating-point multiplier float\_mu1)
3. **CORDIC Iteration Engine (IEEE-754 Arithmetic)**
   * FSM runs up to n=20 iterations
   * Uses **floating-point adders, subtractors, multipliers, and shifters**
   * Updates:
     + xi+1 = xi − di \* yi \* 2⁻ᶦ
     + yi+1 = yi + di \* xi \* 2⁻ᶦ
     + zi+1 = zi − di \* alpha[i]
   * Direction di chosen by sign of zi
   * Hyperbolic mode: extra iterations at i=4,5,8,9

**Floating Point Units used inside iteration:**

* + **Shift/Scaler** → pre-stored 2⁻ᶦ constants (IEEE-754)
  + **Adder/Subtractor (addsubs)** → IEEE-754 addition/subtraction
  + **Multiplier (float\_mu1, multi3)** → IEEE-754 multiplication
  + **Divider (div1)** → IEEE-754 division

1. **Result Computation**
   * After iterations:
     + xi ≈ cos(θ) / cosh(θ) (IEEE-754 float)
     + yi ≈ sin(θ) / sinh(θ) (IEEE-754 float)
2. **Derived Function Unit (Floating Point Ratios)**
   * tan = sin/cos (IEEE-754 divider)
   * cot = cos/sin
   * sec = 1/cos
   * cosec = 1/sin
3. **Control FSM (State Machine)**
   * Controls sequencing of initialization, iteration, and output computation
   * A screenshot of a computer

     AI-generated content may be incorrect.States: Init → Iteration (s1–s6) → Function Result (s13–s14) → Ratios (s7–s12) → Done

Floating Point Based Implementation – Required Hardware is Less

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Total computations Time=885nsec=88clk ;1clk=10nsec