
CORDIC Functions CDDP & CDPS

Introduction

The two HammerCores by Altera® CORDIC macros, CDDP and CDPS, use the CORDIC algorithm to convert rectangular to polar coordinates. This algorithm is also known as the backwards rotation CORDIC algorithm. The CORDIC algorithm is an iterative process, which is often more efficient to implement in hardware than calculating the hypotenuse and angle using multiplies, divides, and square roots.

The accuracy of the CORDIC algorithm can be quite good, depending on the precision of the inputs. The number of steps (iterations) is the same as the precision of the input.

The CDDP macro is a parallel implementation of the CORDIC algorithm, and can be purely combinatorial, or pipelined at every iteration. The CDPS macro is a serial implementation, requiring as many clock cycles as the precision of the inputs to compute the result.

Macro Implementation

The macros accept two unsigned numbers, giving the (x,y) coordinates of the endpoint of the vector. If the location of the point is not in the first quadrant, the absolute values of the endpoint must be input. This will not affect the magnitude output, as the length of the vector is always a positive number.

The backwards rotation CORDIC algorithm only converges in the range of +1 to -1 radians, and the CDDP and CDPS macros are further restricted to convergence in the range of 0 to 1 radians. Converging outside of this range can be easily handled external to the macro. If the y-value is greater than the x-value, the two inputs are switched. This has the effect of making the macro calculate the angle between the y-axis and the vector. The angle between the vector and the x-axis is 90° minus the result, or in radians, (1.571 – result).

More details are given under the applications heading.

There are some very valid reasons why the macros only operate on unsigned numbers. The function scales geometrically in resource requirements, with precision. Converting from signed to unsigned numbers external to the macros is much more resource efficient than operating on signed numbers internal to the macros. As the CORDIC algorithm will only converge in the range of +1 to -1 radians, the inputs will have to be checked, and possibly swapped, to ensure correct results. Angles of vectors outside of the first quadrant can also be calculated at this stage.

One effect of the CORDIC algorithm is that the output magnitude will be scaled. The magnitude of the vector will be about 1.16 times the size of the result calculated by the Pythagorean theorem. The exact value of the scaling will depend on the precision of the calculation, but will always be constant for a given word-length. To calculate the exact scaling value, the macros can be tested by entering a full scale (x,y) pair, and comparing the result to the expected (Pythagorean) result.

Parameters & Ports

CDPF - Parameters & Ports

<i>Table 1. CDPF Parameters</i>	
Parameter	Description
INWIDTH	This is the precision of the input cartesian coordinates (x,y). The valid range is from 6 to 24. INWIDTH is also the precision of the ANGLE[] result. The precision of the vector length will be INWIDTH + 2.
PIPELINED	When this parameter is "YES", the macro will be pipelined with a total of INWIDTH stages. When it is 'NO', the macro will be purely combinatorial.

<i>Table 2. Input Signals</i>	
Parameter	Description
SYSCLK	The SYSCLK input is optional, and is used only if the PIPELINED parameter is "YES". When it is used the result will appear at the output ports INWIDTH clock cycles after the input.
XX[inwidth..1]	This input is for the horizontal, or X-axis, input. The input is unsigned.
YY[inwidth..1]	This input is for the vertical, or Y-axis, input. The input is unsigned.

<i>Table 3. Output Signals</i>	
Parameter	Description
RADIUS[precision..1]	This output bus contains the magnitude of the input positions. This result is scaled. The precision value is INWIDTH + 2, to allow for word growth during the calculation of the radius.
ANGLE[inwidth..1]	This bus contains the angle, in radians, of the vector from the origin to the input point. The value is unsigned, as the endpoint of the vector is unsigned (first quadrant only). The angle is represented as a ratio of the maximum value, which is taken as 1 radian. Ex: with 8 bit precision, 1 radian is represented by 256. 45 degrees, or 0.7854 radians, would be represented as 201.
YY[inwidth..1]	This input is for the vertical, or Y-axis, input. The input is unsigned.

CDPS - Parameters & Ports

Table 4. CDDP Parameters

Parameter	Description
INWIDTH	This is the precision of the input cartesian coordinates (x,y). The valid range is from 6 to 24. INWIDTH is also the precision of the ANGLE[] result. The precision of the vector length will be INWIDTH + 2.

Table 5. Input Signals

Parameter	Description
RESET	RESET is active high, and is required before processing each (x,y) input pair.
SYSCLK	The SYSCLK is used to operate the serial macro. After RESET, the result appears at the output ports INWIDTH clock cycles after the input.
XX[inwidth..1]	This input is for the horizontal, or X-axis, input. The input is unsigned.
YY[inwidth..1]	This input is for the vertical, or Y-axis, input. The input is unsigned.

Table 6. Output Signals

Parameter	Description
RADIUS[precision..1]	This output bus contains the magnitude of the input positions. This result is scaled. The precision value is INWIDTH + 2, to allow for word growth during the calculation of the radius.
ANGLE[inwidth..1]	This bus contains the angle, in radians, of the vector from the origin to the input point. The value is unsigned, as the endpoint of the vector is unsigned (first quadrant only). The angle is represented as a ratio of the maximum value, which is taken as 1 radian. e.g.: with 8-bit precision, 1 radian is represented by 256. 45 degrees, or 0.7854 radians, would be represented as 201.

Examples**CDPP Examples**

Pipelining has no effect on the size of the CDPP macro, only on the throughput. The number of pipeline stages is the same as the input precision of the macro. A new result is computed with each clock cycle.

The examples in Table 7 were compiled into Altera EPF10K10A-1 devices.

<i>TABLE 7. CDPP Examples</i>		
Precision	Size (LCs)	Performance (MHz)
8	290	72
12	634	58
16	1109	53

CDPS Examples

The CDPS macro operates at approximately the same maximum clock frequency as the CDPP macro, but the throughput is lower, as a new result requires as many clock cycles to calculate as the input precision.

The examples in Table 8 were compiled into Altera EPF10K10A-1 devices.

<i>TABLE 8. CDPS Examples</i>			
Precision	Size (LCs)	Performance (MHz)	Throughput (MHz)
8	160	57	7.2
12	239	53	4.4
16	334	47	3.0

Example Calculations

Example calculations will be the same for both macros.

Example 1:

8 bit inputs, (x,y) = (100,100).

Expected result : Magnitude – $1.16 \cdot 141 = 164$, Angle - 45° , or $.785 \text{ rad} = .785 \cdot 256 = 201$

Actual result : Magnitude - 166, Angle – 199

Example 2:

8 bit inputs, $(x,y) = (253,154)$.

Expected result : Magnitude – $1.16*296 = 343$, Angle – 31.3° , or $.545 \text{ rad} = .545*256 = 139$

Actual result : Magnitude - 346, Angle – 139

Example 3:

8 bit inputs, $(x,y) = (222,47)$.

Expected result : Magnitude – $1.16*229 = 263$, Angle – 12° , or $.21 \text{ rad} = .21*256 = 53$

Actual result : Magnitude - 264, Angle – 49

Quadrant Operation**Applications**

This section will explain some techniques for handling any (x,y) point in any quadrant. (These functions have also been integrated into a completely parameterized package, in the Hammercores by Altera arctan macro).

Handling the magnitude is very simple. As the magnitude of the vector is always a positive number, no special handling is required. The absolute values of the (x,y) point are taken, and entered into the macro.

The quadrant of the angle can be determined by the sign of the x and y components.

x positive, y positive	: 1 st quadrant
x negative, y positive	: 2 nd quadrant
x negative, y negative	: 3 rd quadrant
x positive, y negative	: 4 th quadrant

As the CORDIC algorithm will only converge for +1 to -1 radians, the values of (x,y) must be such that the angle is less than 1 radian. If the value of y is greater than the value of x (i.e. the angle is greater than 0.7854 radians), the values of x and y should be interchanged. The angle produced by the macro will then be $\pi/2$ - angle.

To summarize, the following steps should be taken to produce the correct angle:

1. Determine the quadrant of the vector, by looking at the sign of the x and y components.
2. Take the absolute value of the x and y components.
3. If $(y > x)$, interchange (x,y) .
4. Feed (x,y) into the macro.
5. The RADIUS [] output will always contain the correct value.

6. If x and y were interchanged, subtract the ANGLE [] from $\pi/2$.
7. Add the angle offset for the quadrant the vector was into the angle.
 - for the 1st quadrant, do nothing
 - for the 2nd quadrant, add $\pi/2$
 - for the 3rd quadrant, add π
 - for the 4th quadrant, and $3/2\pi$

Example 1 – 8 bit CDDP or CDPS (9 bit inputs)

(x,y) = (100,50)

1. Both numbers are positive, quadrant 1.
2. Absolute values are (100,50).
3. (x > y), do not interchange
4. Feed (100,50) into the macro.
5. RADIUS [] = 134.
6. (x,y) not interchanged, ANGLE [] = 121 (121/256 = 0.473 radians)
7. (x,y) was in first quadrant, ANGLE [] = 121.

Example 2 – 8 bit CDDP or CDPS (9 bit inputs)

(x,y) = (-50,-100)

1. Both numbers are negative, quadrant 3.
2. Absolute values are (50,100).
3. (y > x), interchange
4. Feed (100,50) into the macro.
5. **RADIUS**[] = 134.
6. (x,y) interchanged, **ANGLE**[] = 402 - 121 = 281.*
7. (x,y) was in third quadrant, **ANGLE**[] = 281 + 804 = 1085 (1085/256 = 4.238 radians). *

* : 90 degrees ($\pi/2$) = 256 * $\pi/2$ = 402. 180 degrees ($\pi/2$ = start of quadrant 3) = 2*402 = 804.



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>

Copyright © 2000 Altera Corporation. Altera, FLEX, and AMPP are trademarks and/or service marks of Altera Corporation in the United States and other countries. Other brands or products are trademarks of their respective holders. The specifications contained herein are subject to change without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services. All rights reserved.

