OpenCores
www.opencores.org

Trigonometric functions (degrees) in double FPU

Author: [M.ADITYA]
[muni_aditya@yahoo.com]

Rev. [0.1]
June 19, 2013

This page has been intentionally left blank.

Revision History

Rev.	Date	Author	Description
0.1	04/06/13	M.ADITYA	It takes unsigned bit as input and gives out the degree value in IEEE-754(double) format i.e double precision floating point format.

Contents

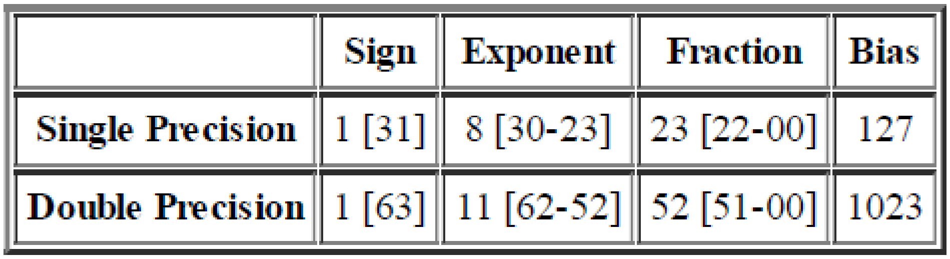
1

Introduction

In computing, floating point describes a method of representing an approximation to real numbers in a way that can support a wide range of values. The numbers are, in general, represented approximately to a fixed number of significant digits (the mantissa) and scaled using an exponent. The base for the scaling is normally 2, 10 or 16. The typical number that can be represented exactly is of the form:

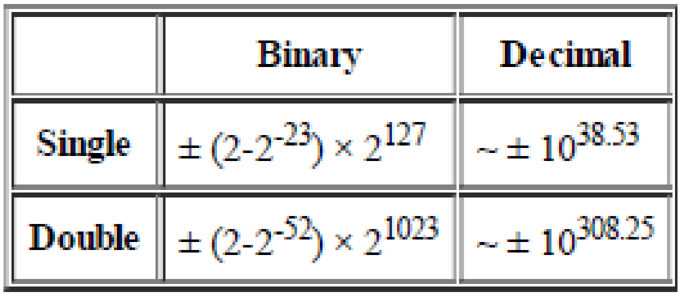
$$\textit{Significant digits} \times \textit{base}^{\,\textit{exponent}}$$

The term floating point refers to the fact that their radix point (decimal point, or, more commonly in computers, binary point) can "float"; that is, it can be placed anywhere relative to the significant digits of the number. This position is indicated as the exponent component in the internal representation, and floating-point can thus be thought of as a computer realization of scientific notation. Over the years, a variety of floating-point representations have been used in computers. However, since the 1990s, the most commonly encountered representation is that defined by the IEEE 754 Standard.



	Sign	Exponent	Fraction	Bias
Single Precision	1 [31]	8 [30-23]	23 [22-00]	127
Double Precision	1 [63]	11 [62-52]	52 [51-00]	1023

Ranges of floating point units:



	Binary	Decimal
Single	$\pm (2-2^{-23}) \times 2^{127}$	$\sim \pm 10^{38.53}$
Double	$\pm (2-2^{-52}) \times 2^{1023}$	$\sim \pm 10^{308.25}$

Double Precision Floating Point Numbers

The IEEE 754 standard defines how double precision floating point number are represented. 64 bits are used to represent a double precision floating point number.

Sign	Exponent	Mantissa
63	62……….52	51………………………………………………………….0

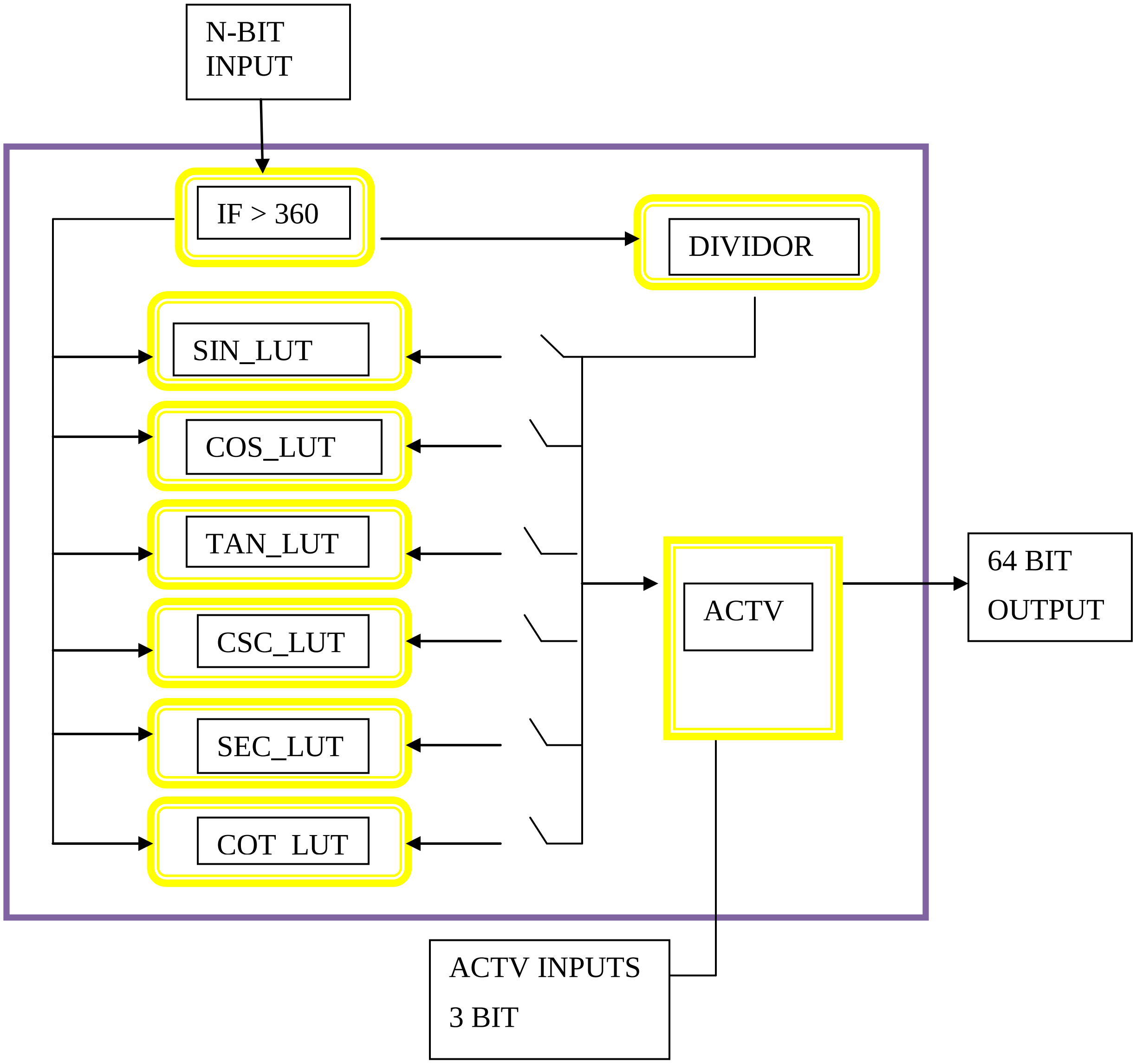
The sign bit occupies bit 63. ‘1’ signifies a negative number, and ‘0’ is a positive number. The exponent field is 11 bits long, occupying bits 62-52. The value in this 11-bit field is offset by 1023, so the actual exponent used to calculate the value of the number is 2^(e-1023). The mantissa is 52 bits long and occupies bits 51-0. There is a leading ‘1’ that is not included in the mantissa, but it is part of the value of the number for all double precision floating point numbers with a value in the exponent field greater than 0.

Value = -1^(sign bit) * 2^(exponent – 1023) * 1.(mantissa)

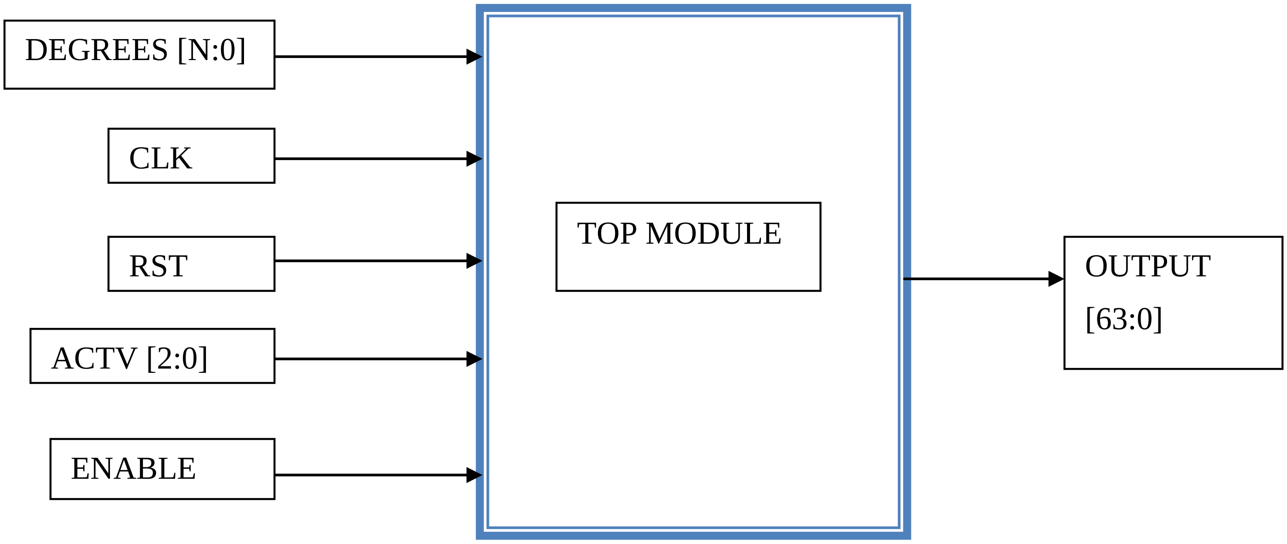
2

Architecture

BLOCK DIAGRAM HIERARCHY)



BLOCK DIAGRAM (TOP LEVEL)



The input signals to the top level module are the following:

1. clk (global)

2. rst (global)

3. enable (set high to start operation)

4. actv (activation, 3 bits, 000 = sin_enable, 001 = cos_enable, 010 = tan_enable, 011 = csc_enable, 100 = sec_enable, 101 = cot_enable)

5. Degrees (input operands can be changed to any number of bits, in this 32 bits)

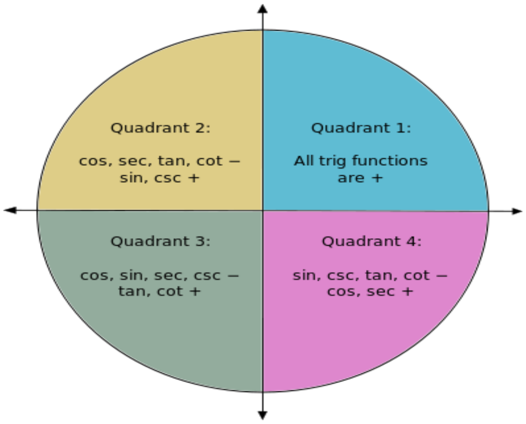
6. data1 (output from operation, 64 bits)

3

Operation

In this version all the trigonometric modules are created as look up table (LUT). To all the input values there is an equivalent double precision floating point unit value, to the input an un-signed value is given. The input port bits can be configured to any number of bits. It also supports all quadrants i.e. sine and cosecant is positive in first and second quadrants and negative in third and fourth quadrants, tangent and cotangent are positive in first and third quadrants and negative in second and fourth quadrants, cosine and secant are positive in first and fourth quadrants and negative in second and third quadrants.

If the value of the degrees is greater than 360 then the value is passed into the divider circuit, in this circuit the total value is divided by the value 360 and returns the remainder value that is less than 360, then the value is given as degrees to the top module and checks for the positive or negative quadrants. "ACTV" value decides which block to activate and passes the corresponding input value to that block and the output value is the same value as in the LUT or changed according to the corresponding quadrants.



This architecture is created using look up table. But instead of creating the table for all the values it is created only for the first 90 values and all the remaining values are derived using these 90 values. From the top module using “ACTV” Input we can choose which function to activate.

Case1:

If the given input value is less than 90 degrees from the top module “00” value is passed on to the “quad” register, this register helps us to know in which quadrant the value lies in. So that it can choose which value to be positive or negative in that particular quadrant and the first bit changed to either 0 or 1 accordingly as required.

Case2:

If the value is between 91 and 180, then the values should be subtracted from the decimal value 180 so that value will be mirrored. For example the value of 89 and 91 would be the same. When the value is in between 90 and 180, 180 is subtracted from input value as the value is less than 180, the resultant will be positive. In the same way all the values can be mirrored using the existing 90 values, “01” value is passed on to “quad” reg.

E.X: if the input is 99 which is between 180 and 90, 180 is subtracted from input value (i.e. 180 – 99) so the resultant value will be 81, the value of any function 99 and 81 will be equal. In this way using only 90 values the remaining values can be derived which are between 90 and 180.

Case3:

If the value is between 181 and 360, then the value should be subtracted from 180. Again in this two things should be considered, after subtracting the input from 180 if the value is less than 90 then the case 1 is repeated “10” value is passed on to quad, else if the resultant is greater than 90 then the case 2 is repeated, “11” value is passed on to the “quad” register.

E.X: if the value is 210,then the input value is subtracted from 180 (i.e. 210 – 180), the resultant is 30 which is less than 90 so the output will be 30 degrees value of the function. The value of any function 30 and 210 are equal.

E.X. if the value is 299, then the input value is subtracted from 180 (i.e. 299 – 180), the resultant is 119 which is greater than 90, so 180 is subtracted from the resultant (i.e. 180 – 119), the resultant is 61. The output value of any function 61 and 299 are equal.

Case4:

If the value is greater than 360 then the value is passed on to divider module. Modulo division is performed in this block, this division is performed until the remainder value is less than 360, remainder value is taken and all the above cases will be repeated. The respective “quad” value will be passed on to the respective module.

In each case a “quad” value is passed based on the input degrees value, these “quad” values are passed on to the respective module. Each module consist of the sign value, they will be changed or remain the same according to the respective quadrants.

4

Registers

List of Registers

Name	Address	Width	Description
ACTV	3’b000	3 bit	Activates the sine block
ACTV	3’b001	3 bit	Activates the cosine block
ACTV	3’b010	3 bit	Activates the tangent block
ACTV	3’b011	3 bit	Activates the cosecant block
ACTV	3’b100	3 bit	Activates the secant block
ACTV	3’b101	3 bit	Activates the cotangent block
QUAD	2’b00	2 bit	Specifies the value is in first quadrant
QUAD	2’b01	2 bit	Specifies the value is in second quadrant
QUAD	2’b10	2 bit	Specifies the value is in third quadrant
QUAD	2’b11	2 bit	Specifies the value is in fourth quadrant
DATA1	output	Bit	Gives the final result

Table 1: List of registers

Register DATA1 – Description

Bit	Access	Description
63	sign	It specifies whether the value is positive or negative
62-52	exponent	These bits store the value of the exponent that is subtracted from the bias
51-0	mantissa	It consist of the fractional value of the number

5

IO ports

Port	Width	Direction	Description
CLK	1 bit	Input	Clock input, common for all blocks
RST	1 bit	Input	Resets all the values
ACTV	4 bits	Input	Activates the selected block
ENABLE	1bit	Input	Enables top block and a particular block
Input	N bits	Input	Can be configured to any number of bits
Data1	64 bits	Output	Gives the 64 bit output result in double precision floating point format