



Floating Point Natural Logarithm (ALTFP_LOG)

Megafunction User Guide



101 Innovation Drive
San Jose, CA 95134
www.altera.com

Software Version: 8.1
Document Version: 1.0
Document Date: November 2008

Copyright © 2008 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time 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.

Chapter 1. About this Megafunction

Introduction	1-1
Background	1-1
Floating Point Natural Logarithm Megafunction (ALTFP_LOG) and Features	1-2
General Description	1-2

Chapter 2. MegaWizard Plug-In Options for ALTFP_LOG Megafunction

Introduction	2-1
Design Example	2-5
Configuration Settings	2-5
Functional Simulation in the ModelSim-Altera Software	2-6
Understanding the Simulation Results	2-6

Chapter 3. Specifications

Ports and Parameters	3-1
----------------------------	-----

Additional Information

Revision History	Info-1
Referenced Document	Info-1
How to Contact Altera	Info-1
Typographic Conventions	Info-2

Introduction

As FPGAs are increasingly being used for floating point computing, Altera provides the parameterizable floating point megafunctions such as the ALTFP_ADD_SUB, ALTFP_DIV, and ALTFP_MULT that are complied with the IEEE 754 Floating Point Standard. Beside these basic megafunctions, FPGAs require good implementations of other functions such as logarithm, exponential, and so on, for complex floating point computations.

The ALTFP_LOG megafunction is one of the floating point units that are being offered by Altera towards a complete solution of floating point megafunctions. Altera provides the parameterizable megafunctions that are optimized for Altera® device architectures. Using megafunctions instead of coding your own logic saves valuable design time.

Background

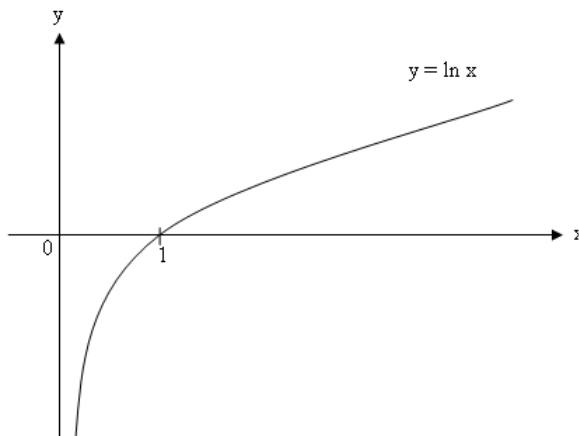
A logarithm with base b of a number N , $\log_b N$, is equal to the power to which the base b would have to be raised to be equal to the number N . For example, if $\log_b N = 2$, then $N = b^2$.

The natural logarithm, formerly known as the hyperbolic logarithm, is a special logarithm with base e , where e is an irrational constant approximately equal to 2.718. Therefore, the natural logarithm of a number N , $\log_e N$, simply means the power to which the base e would have to be raised to equal the number N . In terms of notation, natural logarithm is normally represented by \ln instead of \log_e .

The natural logarithm can only be defined for positive real numbers. The natural logarithm of zero or any negative number is undefined.

Figure 1-1 shows the graph of a natural logarithm function, $y = \ln x$.

Figure 1-1. Graph for a Natural Logarithm Function



The natural logarithm function has the following characteristics:

- When limit $x \rightarrow 0, y \rightarrow -\infty$
- For $0 < x < 1, y < 0$ (negative)
- When limit $x \rightarrow 1, y \rightarrow 0$
- For $1 < x < e, 0 < y < 1$
- When limit $x \rightarrow e, y \rightarrow 1$
- When limit $x \rightarrow \infty, y \rightarrow \infty$

In mathematics, the natural logarithm can be used to calculate the area under the reciprocal function. There are also many practical applications in the fields of earthquake measurement, electronics, mechanics, and finance that use the natural logarithm function. An example of the natural logarithm's application in the field of finance is to calculate the compounds period n needed to double the principle for a given nominal interest rate i that is expressed as a decimal.

Equation 1-1 shows the calculation for the compounds period n :

Equation 1-1. Calculation for Compounds Period n Needed to Double the Principle for a Given Nominal Interest Rate i

$$n = \frac{\ln 2}{\ln(1 + i)}$$

Floating Point Natural Logarithm Megafunction (ALTFP_LOG) and Features

The ALTFP_LOG megafunction implements natural logarithm functions and offers the following features:

- Single precision, double precision, and single-extended precision support
- Input support for normal numbers, infinity, zero, denormal numbers, and not-a-number (NaN)
- Two status output ports: `zero` and `nan`
- Optional input ports, including asynchronous clear (`aclr`) and clock enable (`clk_en`)

General Description

This section discusses the IEEE 754 Floating Point Standard, the special case numbers handling, the exception handling, the rounding, and the pipeline requirement for the ALTFP_LOG megafunction.

Supported Format

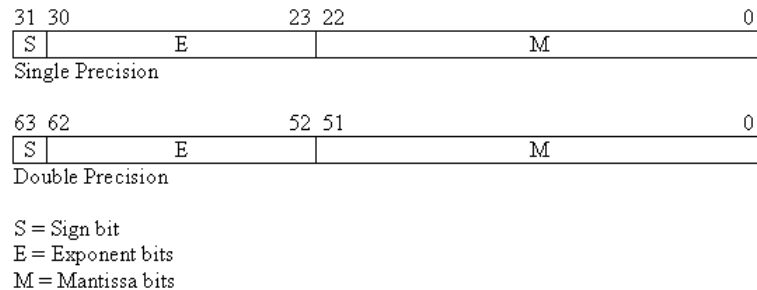
The IEEE 754 Floating Point Standard defines four formats for floating point numbers—single precision, double precision, single extended precision, and double extended precision. The ALTFP_LOG megafunction only supports three formats: single precision, double precision, and single extended precision.

In single precision format, there is a 1-bit Sign, 8-bit Exponent, and 23-bit Mantissa (an implicit 24th bit is the leading 1 in all Mantissas, which is not stored), that result in a 32-bit number.

In double precision format, there is a 1-bit Sign, 11-bit Exponent, and 52-bit Mantissa (an implicit 53rd bit is the leading 1 in all Mantissas, which is not stored), that result in a 64-bit number.

Figure 1–2 shows the bit patterns of the IEEE 754 floating point number representation for single precision and double precision, respectively.

Figure 1–2. IEEE 754 Floating Point Format



The 1-bit Sign indicates whether the number is a positive number or a negative number. Zero (0) denotes a positive number, and one (1) denotes a negative number.

The exponent field represents both positive and negative exponents. Therefore, a bias is added to the actual exponent in order to get the stored exponent. Conversely, a bias needs to be subtracted from the stored exponent to get its actual exponent.

Equation 1–2 shows how the bias for certain precision format is calculated.

Equation 1–2. Calculation for the Value of Bias

$$Bias, B = 2^{(E-1)} - 1$$

In Equation 1–2, *E* is equal to the number of bits in the exponent field. For example, the single precision number has *E* = 8, and therefore the bias is 127.

Therefore, in single precision format, the actual exponent needs to be added to 127 before being stored in the exponent field. Conversely, to get the actual exponent for the purpose of computation, 127 needs to be subtracted from the stored exponent.

Table 1–1 summarizes the data bit(s) of the sign, the exponent, and mantissa fields for floating point numbers, the bias value, and the valid range of the stored exponent and actual exponent for different precisions supported by the ALTFP_LOG megafunction.

Table 1-1. Summary for Different Precisions Supported by ALTFP_LOG Megafunction

Precision	Sign Bit (S)	Exponent Bits (E)	Mantissa Bits (M)	Total Bits (T)	Bias (B)	Valid Range of Stored Exponent	Valid Range of Actual Exponent (1)
Single	1	8	23	32	127	1 to 254 (2)	-126 to 127
Double	1	11	52	64	1023	1 to 2046 (3)	-1022 to 1023
Single Extended	1	$11 \leq E < M$	$31 \leq M \leq 52$	$43 \leq T \leq 64$	$2^{(E-1)} - 1$	1 to 2^{E-2} (4)	$2 \cdot 2^{(E-1)}$ to $2^{(E-1)} - 1$

Notes to Table 1-1:

- (1) To get the actual exponent, subtract the bias from the stored exponent.
- (2) In single precision format, the stored exponent of 0 (all 0's) and 255 (all 1's) are used to represent special cases.
- (3) In double precision format, the stored exponent of 0 (all 0's) and 2047 (all 1's) are used to represent special cases.
- (4) In single extended precision format, the stored exponent of 0 (all 0's) and $2^E - 1$ (all 1's) are used to represent special cases.

Special Case Numbers

The IEEE 754 Floating Point Standards divides the special case numbers into a few categories. The categories for the special case numbers are zero, positive denormalized, negative denormalized, positive infinity, negative infinity, and not-a-number (NaN).

Table 1-2 shows the data bits for special case IEEE 754 floating point numbers.

Table 1-2. Data Bits for Special Case IEEE 754 Floating Point Numbers

Special Case	Sign Field	Exponent Field	Mantissa Field
Zero	Don't care	All 0's	All 0's
Positive Denormalized	0	All 0's	Non-zero
Negative Denormalized	1	All 0's	Non-zero
Positive Infinity	0	All 1's	All 0's
Negative Infinity	1	All 1's	All 0's
NaN	Don't care	All 1's	Non-zero

The next section discusses the ALTFP_LOG megafunction's exception handling on the input of these special cases.

Exception Handling

Only positive numbers are valid for the operation of natural logarithms. Therefore, the ALTFP_LOG megafunction only produces valid output if the input is a positive number. The input of negative number results in a NaN output and asserts the nan flag signal (if it is used) of the megafunction. For positive input numbers, the megafunction has different exception handlings for different special cases of input values.


Table 1-3 shows the output exception handlings of the megafunction for the special cases of input values.

Table 1-3. Output Exception Handlings for Special Cases of Input Values

Input		Output	
Sign Bit	Special Cases	Special Cases	Flag Signal(s) if Applicable
1 (negative)	Any Values	NaN (1)	nan = HIGH
0 (positive)	Zero (2)	Negative Infinity	—
	Positive Denormalized (3)	Negative Infinity	—
	Positive Infinity	Positive Infinity	—
	NaN	NaN	nan = HIGH
	1 (4)	Zero	zero = HIGH

Notes to Table 1-3:

- (1) The natural logarithm of a negative value is invalid. Therefore, the ALTFP_LOG megafunction produces NaN type of output. For NaN output, the megafunction outputs the exponent field as to All 1's and the mantissa field as to non-zero. The megafunction uses 100...00 (all 0's except the MSB equal to 1) to represent non-zero output in mantissa field.
- (2) The zero in this case represents zero special case of the IEEE standard. It does not equivalent to ln 0, but instead it approximates to it.
- (3) The value of positive denormalized special case is a value approximates to zero, and the natural logarithm of a value approximates to zero produces output of negative infinity.
- (4) The "1" in this case is equivalent to ln 1, in which it is the actual value, not the input stored data in the IEEE 754 Floating Point format. In the IEEE 754 Floating Point format, the actual value "1" is represented by all 1's except the MSB for the exponent field, and all 0's for the mantissa field.

 For definition of the special cases, refer to [Table 1-2 on page 1-4](#).

Rounding

In the IEEE-754 standard, there are four types of rounding modes: round-to-nearest-even, round-toward-zero, round-toward-positive-infinity, and round-toward-negative-infinity. The most commonly used rounding mode is round-to-nearest-even. This megafunction supports only round-to-nearest-even mode. With round-to-nearest-even, the result is rounded to the nearest floating-point number. If the result is exactly halfway between two floating-point numbers, it is rounded so that the LSB becomes a zero, which is even.

Pipeline Requirement

The number of the pipeline required for the ALTFP_LOG megafunction is fixed and depends on the precision used, the width of the mantissa, or both. The number of the pipeline reflects the output latency in clock cycles.

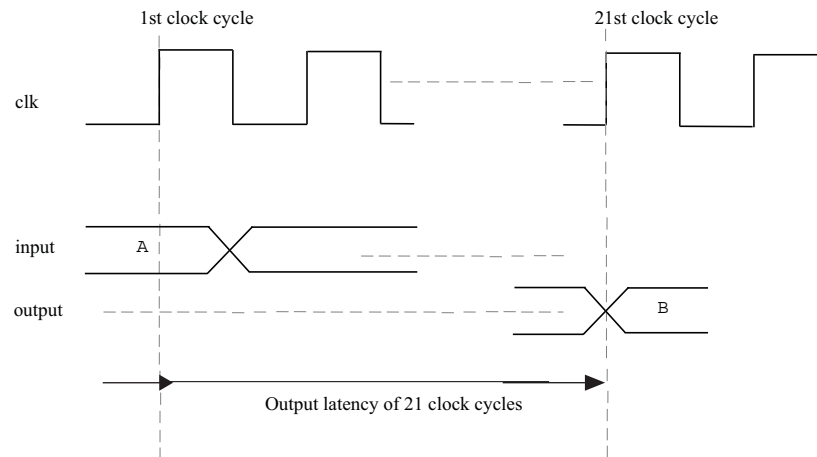
[Table 1-4](#) shows the pipeline requirement for each precision format.

Table 1-4. Pipeline Requirement over Precision

Precision	Mantissa Width	Number of Pipeline
Single	23	21
Double	52	34
Single Extended	31-36	25
	37-42	28
	43-48	31
	49-52	34

Figure 1-3 shows an example of the pipeline effect on the single precision mode of the ALTFP_LOG megafunction. It illustrates the latency of output B corresponding to input A. That is, $B = \ln A$.

Figure 1-3. Output Latency for ALTFP_LOG in Single Precision Mode



The output shows an undefined value before the output latency is reached. To get the valid value, sample the output only after the respective output latency is reached.

Performance Targeting Stratix III Devices

You can get the performance information under timing report after compiling your design with the Quartus® II software.

Table 1-5 provides an approximation of performance targeting Stratix® III devices.

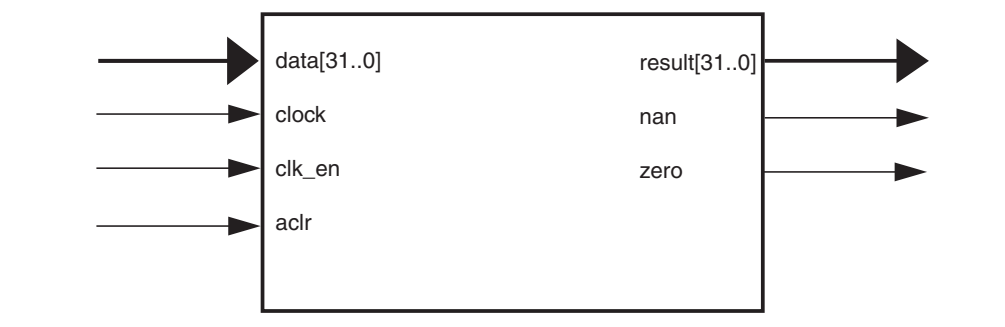
Table 1-5. Approximation of Performance Targeting Stratix III Devices

Precision	f_{MAX} (MHz)	Resource Usage		
		ALUTs	Registers	18-bit DSP
Single	360	1700	2400	8
Double	170	5000	7800	64

Introduction

The ALTFP_LOG megafunction contains four input ports and three output ports. [Figure 2-1](#) shows the ports for the ALTFP_LOG megafunction. The compulsory ports are `data`, `clock` and `result`; other ports are optional. The ALTFP_LOG megafunction can be easily configured and built through the MegaWizard® Plug-In Manager.

Figure 2-1. Ports for ALTFP_LOG Megafunction



The Quartus II software provides the MegaWizard Plug-In Manager for you to easily configure and build the megafunction of your choice. It contains a list of plug-ins for different megafunctions. The plug-in is a GUI that guides you through the available options and features to configure the megafunction. The plug-in for the ALTFP_LOG megafunction is listed under the Arithmetic module.

[Table 2-1](#) shows the descriptions for the options available in the ALTFP_LOG MegaWizard Plug-In.

Table 2-1. ALTFP_LOG MegaWizard Plug-In Manager Page Options and Descriptions (Part 1 of 4)

MegaWizard Plug-In Page	Configuration or Option Setting	Descriptions
1	Which action do you want to perform?	You can select from the following options: Create a new custom megafunction variation , Edit an existing custom megafunction variation , or Copy an existing custom megafunction variation .
2a	Select a megafunction from the list below	Select ALTFP_LOG from the Arithmetic category.
	Which device family will you be using?	Specify the device family that you want to use.
	Which type of output file do you want to create?	You can choose from AHDL (.tdf), VHDL (.vhd), or Verilog HDL (.v) as the output file type.
	What name do you want for the output file?	Specify the name of the output file without the file extension.
	Return to this page for another create operation	Turn on this option if you want to return to this page to create another megafunction.

Table 2-1. ALTFP_LOG MegaWizard Plug-In Manager Page Options and Descriptions (Part 2 of 4)

MegaWizard Plug-In Page	Configuration or Option Setting	Descriptions
3	Currently selected device family	Specifies the device family you chose on page 2a.
	Match project/default	Turn off Match project/default if you want to select a device family that is different from the device family you chose in page 2a. Turn on this option if you want to use the device family you chose in page 2a.
	What is the floating point format?	Choose one of the supported floating point formats: Single precision (32 bits) , Double precision (64 bits) , or Single extended precision (43..64 bits) . (1)
	What is the width of the 'data' input and 'result' output buses?	Floating point input data and output result. The MSB is the sign bit, the next most significant bits are the exponent, and the mantissa occupies the least significant bits. The size of this port is the total width of sign bit, exponent bits, and mantissa bits. For single precision, it is fixed to 32 bits. For double precision, it is fixed to 64 bits. For single extended precision, you can choose from 43 bits to 64 bits. (2)
	What is the width of the exponent port?	For single precision, it is fixed to 8 bits. For double precision, it is fixed to 11 bits. For single extended precision, it ranges from 11 bits to 30 bits, depending on the data width. (2)
	Mantissa width =(data width)-(exponent width)-1	For single precision, it is fixed to 23 bits. For double precision, it is fixed to 52 bits. For single extended precision, it ranges from 31 bits to 52 bits, depending on the data width and exponent width. (2)
	Output latency in clock cycles	The latency is fixed and depends on the pipeline requirement over the precision and mantissa width. (3)

Table 2-1. ALTFP_LOG MegaWizard Plug-In Manager Page Options and Descriptions (Part 3 of 4)

MegaWizard Plug-In Page	Configuration or Option Setting	Descriptions
4	Create an asynchronous clear port	This is an optional <code>aclr</code> input port. When asserted (HIGH), it asynchronously clears the <code>data</code> input port and all the output ports.
	Create a clock enable port	This is an optional <code>clk_en</code> input port. When asserted (HIGH), it enables the <code>data</code> input port and all the output ports. When de-asserted (LOW), no operation is carried out and the output is unchanged. If an operation is in process, it halts the current process until the signal is asserted again. (4) If this port is not created, the <code>data</code> input port and all the output ports are enabled by default and depend on the <code>clock</code> input port.
	Create a 'NaN' port	This is an optional <code>nan</code> output port. It is asserted (HIGH) if the data input is a negative value, or if the data input is a NaN special case. (5)
	Create a 'zero' port	This is an optional zero output port. It is asserted (HIGH) if the result output port shows <code>x0000 0000</code> (zero special case). For single precision, for example, only an input of <code>x3f80 0000</code> (actual value = 1) produces this result, because <code>ln 1</code> is equivalent to 0. (5)

Table 2-1. ALTFP_LOG MegaWizard Plug-In Manager Page Options and Descriptions (Part 4 of 4)

MegaWizard Plug-In Page	Configuration or Option Setting	Descriptions
5	Simulation Libraries	To properly simulate the generated design files for the ALTFP_LOG megafunction in third-party tools, Altera megafunction simulation library (altera_mf) and LPM megafunction simulation library (220model) files are needed.
	Generate netlist	Turn on this option if you want to generate a netlist for your third-party EDA synthesis tool to estimate the timing and resource usage of the megafunction. If you enable this option, a netlist file (_syn.v) is generated. This file is a representation of the customized logic used in the Quartus II software and provides the connectivity of the architectural elements in the megafunction but may not represent true functionality.
6	Summary Page	<p>Specify the types of files to be generated. The Variation file (<i><function name>.v</i>) contains wrapper code in the language you specified on page 2a and is automatically generated. Choose from the following types of files:</p> <ul style="list-style-type: none"> ■ AHDL Include file (<i><function name>.inc</i>) ■ VHDL component declaration file (<i><function name>.cmp</i>) ■ Quartus II symbol file (<i><function name>.bsf</i>) ■ Instantiation template file (<i><function name>.inst.v</i>) ■ Verilog HDL black-box file (<i><function name>.bb.v</i>) <p>For more information about the wizard-generated files, refer to <i>Quartus II Help</i> or to the <i>Recommended HDL Coding Styles</i> chapter in volume 1 of the <i>Quartus II Handbook</i>.</p>

Notes to Table 2-1:


- (1) [Table 2-2 on page 2-5](#) provides an example for the valid input range and input of special case for single precision format that explicitly shows the input in hexadecimal.
- (2) For the summary of the data bit(s) of sign, exponent, and mantissa fields, the bias value, and valid range of stored and actual exponent for different precision format, refer to [Table 1-1 on page 1-4](#).
- (3) For more information about pipeline requirements over precision, refer to “[Pipeline Requirement](#)” on page 1-5.
- (4) Assert the `clk_en` signal for the number of clock cycles equivalent to the required pipeline (output latency) for the results to be shown at the output.
- (5) For more information about handling output exceptions for special cases of input value, refer to [Table 1-3 on page 1-5](#).

Table 2-2. Valid Input Range and Input of Special Case for Single Precision Format

Input (in hexadecimal)	Descriptions
x0000 0000	Zero (1)
x0000 0001 to x007F FFFF	Positive denormalized
x0080 0000 to x7F7F FFFF	Valid input range: Smallest value: x0080 0000 (actual value= $1.00\dots x2^{-126}$) Largest value: x7F7F FFFF (actual value= $1.11\dots x2^{127}$)
x7F80 0000	Positive infinity
x7F80 0001 to x7FFF FFFF	NaN
x8000 0000 to xFFFF FFFF	Negative input

Note to Table 2-2:

(1) The zero in this case represents zero special case of the IEEE standard. It is not equivalent to $\ln 0$, but instead it approximates to it.

 For more information about handling output exceptions, refer to [Table 1-3 on page 1-5](#) for special cases of input values.

Design Example

This design example provides a pre-configured single precision ALTFP_LOG megafunction (**altfplog.v**) that is built through the ALTFP_LOG MegaWizard Plug-In.

Configuration Settings

[Table 2-3](#) shows the settings of the ALTFP_LOG megafunction.

Table 2-3. ALTFP_LOG Megafunction Settings

MegaWizard Page	Available Options	Configured Settings
3	What is the floating point format	Single precision (32 bits)
	What is the width of the 'data' input and 'result' output buses?	Fixed to 32 bits by the MegaWizard Plug-In Manager
	What is the width of the exponent port?	Fixed to 8 bits by the MegaWizard Plug-In Manager
	Mantissa width = (data width) - (exponent width) - 1	Fixed to 23 bits by the MegaWizard Plug-In Manager
	Output latency in clock cycles	Fixed to 21 by the MegaWizard Plug-In Manager
4	Create an asynchronous clear port	Selected
	Create a clock enable port	Selected
	Create a 'nan' port	Selected
	Create a 'zero' port	Selected

In addition to the design variation file (**altfplog.v**), a testbench (**altfplog.vt**) is created for you to run the functional simulation in the ModelSim®-Altera software.

Functional Simulation in the ModelSim-Altera Software

You should be familiar with the ModelSim-Altera software before trying the design example. If you are unfamiliar with the ModelSim-Altera software, refer to the ModelSim support page on the Altera website (www.altera.com). On the support page, there are links to such topics as installation, usage, and troubleshooting.

Set up and simulate the design in the ModelSim-Altera software by performing the following steps:

1. Unzip the **altfplog.zip** file to any working directory on your PC.
2. Start the ModelSim-Altera software.
3. On the File menu, click **Change Directory**.
4. Select the folder in which you unzipped the files.
5. Click **OK**.
6. On the Tools menu, point to **TCL** and click **Execute Macro**. The **Execute Do File** dialog box appears.
7. Select the **altfplog.do** file and click **Open**. The **altfplog.do** file is a script file to automate all the necessary settings, compiles and simulate the design files, and displays the simulation waveform.
8. Verify the results shown in the Waveform Viewer window.

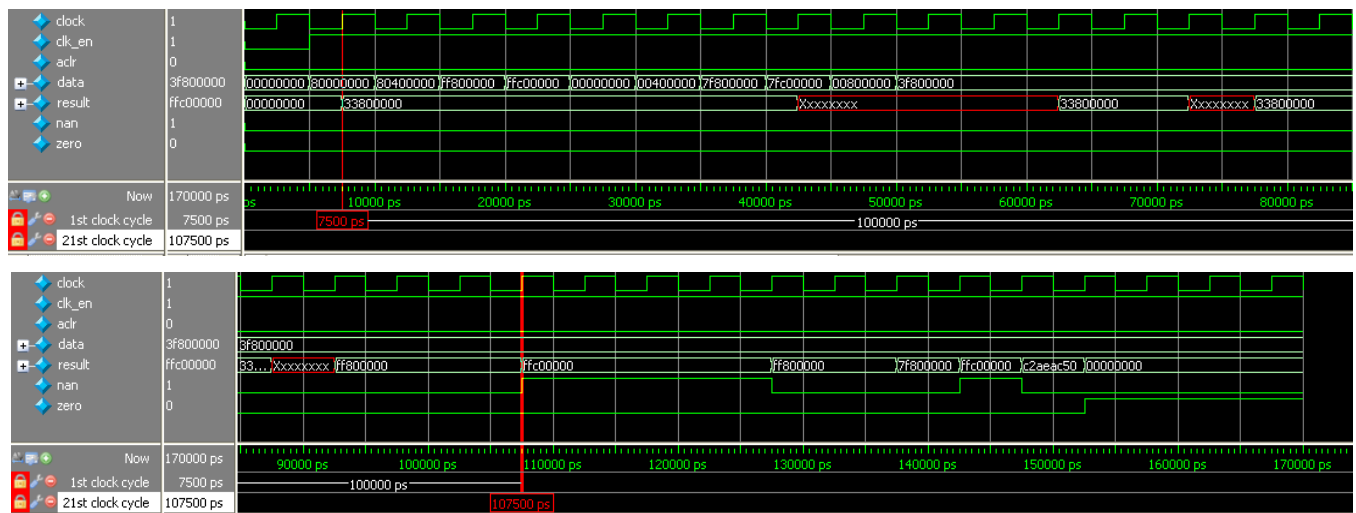
You can rearrange signals, remove signals, add signals, and change the radix by modifying the script in **altfplog.do** accordingly.

Understanding the Simulation Results

This design example includes the input of special cases to show the exception handling of the megafunction, as well as valid inputs such as the smallest valid input and the input value of "1".

[Figure 2-2](#) shows the expected simulation results in the ModelSim-Altera software.

Figure 2-2. Simulation Results



The single precision ALTFP_LOG megafunction requires 21 pipelines. It has an output delay of 21 clock cycles and only shows at the `result` output port at the 21st clock cycle. Therefore, you should only sample the output 21 clock cycles after the input and ignore any output before that.

For output delays of different precision or mantissa width, refer to “Pipeline Requirement” on page 1-5.

If you use the `clk_en` port when it is de-asserted (LOW), no operation is executed or the current operation is halted. The counting of the delay cycle continues when `clk_en` is asserted (HIGH).


After you have run the simulation, you should get a similar output shown in Table 2-4.

Table 2-4. Expected Output for Different Input Vectors

Sign Bit	Exponent Bits	Mantissa Bits	Input Vectors Shown in Hexadecimal	Output (In Hexadecimal)
1 (1)	All 0's	All 0's	x8000 0000	xFFC0 0000
	All 0's	Non-Zero	x8040 0000	
	All 1's	All 0's	xFF80 0000	
	All 1's	Non-Zero	xFFC0 0000	
0	All 0's	All 0's	x0000 0000 (2)	xFF80 0000
	All 0's	Non-Zero	x0040 0000 (3)	
	All 1's	All 0's	x7F80 0000 (4)	x7F80 0000
	All 1's	Non-Zero	x7FC0 0000 (5)	xFFC0 0000
	All 0's Except LSB	All 0's	x0080 0000 (smallest valid input) (6)	xC2AE AC50
	All 1's Except MSB	All 0's	x3F80 0000 (7)	x0000 0000

Notes to Table 2-4:

- (1) Input is negative value. Output is expected to be NaN (xFFC0 0000), and `nan` output port is asserted.
- (2) Zero special case of input. Output is expected to be negative infinity (xFF80 0000).
- (3) Positive denormalized special case of input. Output is expected to be negative infinity (xFF80 0000).
- (4) Positive infinity special case of input. Output is expected to be positive infinity (x7F80 0000).
- (5) NaN special case of input. Output is expected to be NaN (xFFC0 0000), and `nan` output port is asserted.
- (6) The smallest valid input is when all the input bits are 0's except the LSB of the exponent field.
- (7) x3F80 0000 is equivalent to the actual value, $1.0 \times 2^0 = 1$. Since `ln 1` results in zero, it produces output of zero special case, and `zero` output port is asserted.

 For representation of special cases, refer to [Table 1-3 on page 1-5](#).

Ports and Parameters

This chapter details the ports and parameters for the ALTFP_LOG megafunction. [Table 3-1](#) shows the input ports, [Table 3-2](#) shows the output ports, and [Table 3-3](#) shows the parameters of the ALTFP_LOG megafunction. The parameter details are only relevant if you bypass the MegaWizard® Plug-In Manager interface and use the megafunction as a directly parameterized instantiation in your design. The details of these parameters are hidden from the MegaWizard Plug-In Manager interface.

Table 3-1. Input Ports of ALTFP_LOG Megafunction

Port Name	Required	Description	Comments
aclr	No	Asynchronous clear	When the <code>aclr</code> port is asserted high, the input and output ports is asynchronously cleared.
clk_en	No	Clock enable	When the <code>clk_en</code> port is asserted high, a natural logarithm operation takes place. When the signal is asserted low, no operation occurs and the outputs remain unchanged. Deasserting <code>clk_en</code> halts the operation until it is asserted again. Assert the <code>clk_en</code> signal for the number of clock cycles equivalent to the required output latency (PIPELINE parameter value) for the results to be shown at the output.
clock	Yes	Clock input	The <code>clock</code> port is the clock input port to the ALTFP_LOG megafunction.
data[]	Yes	Data input	Floating-point input data. The MSB is the sign, the next most significant bits are the exponent, and the mantissa occupies the least significant bits. This input port size is the total width of the sign bit, exponent bits, and mantissa bits. For single precision, the width is fixed to 32 bits. For double precision, the width is fixed to 64 bits. For single extended precision, you can choose a width in the range from 43 to 64 bits. For the summary of the data bit(s) of sign, exponent, and mantissa fields, the bias value, and valid range of stored and actual exponent for different precision format, refer to Table 1-1 on page 1-4 .

Table 3-2. Output Ports of ALTFP_LOG Megafunction

Port Name	Required	Description	Comments
result []	Yes	The natural logarithm of the value on input data.	The natural logarithm of the data [] at input, shown in floating point format. The width of the result output port and data input port are the same.
zero	No	Optional zero exception output.	Asserted when the exponent and mantissa of the result output port is zero. This occurs when the actual input value is 1 because $\ln 1 = 0$. For more information about handling output exceptions for special cases of input values, refer to Table 1-3 on page 1-5 .
nan	No	Not a Number (NaN) exception output.	Asserted when the exponent and mantissa of the result output port are all 1's and non-zero, respectively. This occurs when the input is a negative number or not a number (NaN). For more information about handling output exceptions for special cases of input values, refer to Table 1-3 on page 1-5 .

Table 3-3. Parameters of ALTFP_LOG Megafunction

Parameter	Type	Required	Comments
WIDTH_EXP	Integer	Yes	Specifies the precision of the exponent. If omitted, the default is 8. The Bias of the exponent is always set to $(2^{(\text{WIDTH_EXP} - 1)} - 1)$, that is, 127 for single precision and 1023 for double precision. The value of WIDTH_EXP must be 8 for single precision, 11 for double precision, and a minimum of 11 for single extended precision. The value of WIDTH_EXP must be less than the value of WIDTH_MAN, and the sum of WIDTH_EXP and WIDTH_MAN must be less than or equals to 64. For more information about the precision of the exponent, refer to Table 1-1 on page 1-4 .
WIDTH_MAN	Integer	Yes	Specifies the precision of the mantissa. If omitted, the default is 23. The value of WIDTH_MAN must be 23 for single precision and 52 for double precision. For single extended precision, the valid value ranges from 31 to 52. The value of WIDTH_MAN must be greater than the value of WIDTH_EXP, and the sum of WIDTH_EXP and WIDTH_MAN must be less than or equals to 64. For more information about the precision of the mantissa, refer to Table 1-1 on page 1-4 .
PIPELINE	Integer	Yes	Specifies the amount of latency, expressed in clock cycles, used in the ALTFP_LOG megafunction. This number is fixed and must be matched with the precision and the width of the mantissa, or both. Create the megafunction with the MegaWizard Plug-in Manager to calculate the value for this parameter, or refer to Table 1-4 on page 1-5 .
DEVICE_FAMILY	String	Yes	Define device family setting. The ALTFP_LOG megafunction is not a family-dependent module, but this parameter is needed. This parameter is used for modeling and behavioral simulation purposes. Create the megafunction with the MegaWizard Plug-In Manager to calculate the value for this parameter.

Revision History

The table below displays the revision history for the chapters in this User Guide.

Date	Document Version	Changes Made
November 2008	1.0	Initial release

Referenced Document

This user guide references the following document:

- *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook*

How to Contact Altera

For the most up-to-date information about Altera products, see the following table.






Contact <i>(Note 1)</i>	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General) (Software Licensing)	Email	nacomp@altera.com
	Email	authorization@altera.com

Note:

- (1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other GUI labels. For example, Save As dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, \qdesigns directory, d: drive, and chiptrip.gdf file.
<i>Italic Type with Initial Capital Letters</i>	Indicates document titles. For example, <i>AN 519: Stratix IV Design Guidelines</i> .
<i>Italic type</i>	Indicates variables. For example, $n + 1$. Variable names are enclosed in angle brackets (<>). For example, <file name> and <project name>. .pof file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
“Subheading Title”	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”
Courier type	Indicates signal, port, register, bit, block, and primitive names. For example, <code>data1</code> , <code>tDi</code> , and <code>input</code> . Active-low signals are denoted by suffix <code>n</code> . For example, <code>resetn</code> . Indicates command line commands and anything that must be typed exactly as it appears. For example, <code>c:\qdesigns\tutorial\chiptrip.gdf</code> . Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword <code>SUBDESIGN</code>), and logic function names (for example, <code>TRI</code>).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
 CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
 WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
	The angled arrow instructs you to press Enter .
	The feet direct you to more information about a particular topic.