Open Floating Point Unit

The Free IP Cores Projects
www.opencores.org

Author: Rudolf Usselmann
rudi@asics.ws
www.asics.ws

Summary:
This documents describes a free single precision floating point unit. This floating point unit can perform add, subtract, multiply, divide, integer to floating point and floating point to integer conversion.

Change Log

13/9/2000 RU
Initial release of FPU

15/9/2000 RU
Added integer to floating point and vise versa conversion

To Do

Things that still need to be done:

- Improve performance by prediction leading zeros in post normalization unit

1. Introduction

This document describes a single precision floating point unit. The floating point unit is fully IEEE 754 compliant.

1.1. The FPU supports the following operations:

fpu_op	Operation
0	Add
1	Subtract
2	Multiply
3	Divide
4	Int to float conversion
5	Float to int conversion
6	Remainder (Future Function)
7	RESERVED

1.2. The FPU supports the following Rounding Modes:

rmode	Rounding Mode
0	Round to nearest even
1	Round to Zero
2	Round to +INF (UP)
3	Round to -INF (DOWN)

1.3. General Operation

The FPU can perform a floating point operation every cycle. It will latch the operation type, rounding mode and operands and deliver a result four cycles later.

The FPU will never generate a SNAN output. The SNAN output is asserted when one of the operands was a signaling NAN (output will be a quiet NAN).

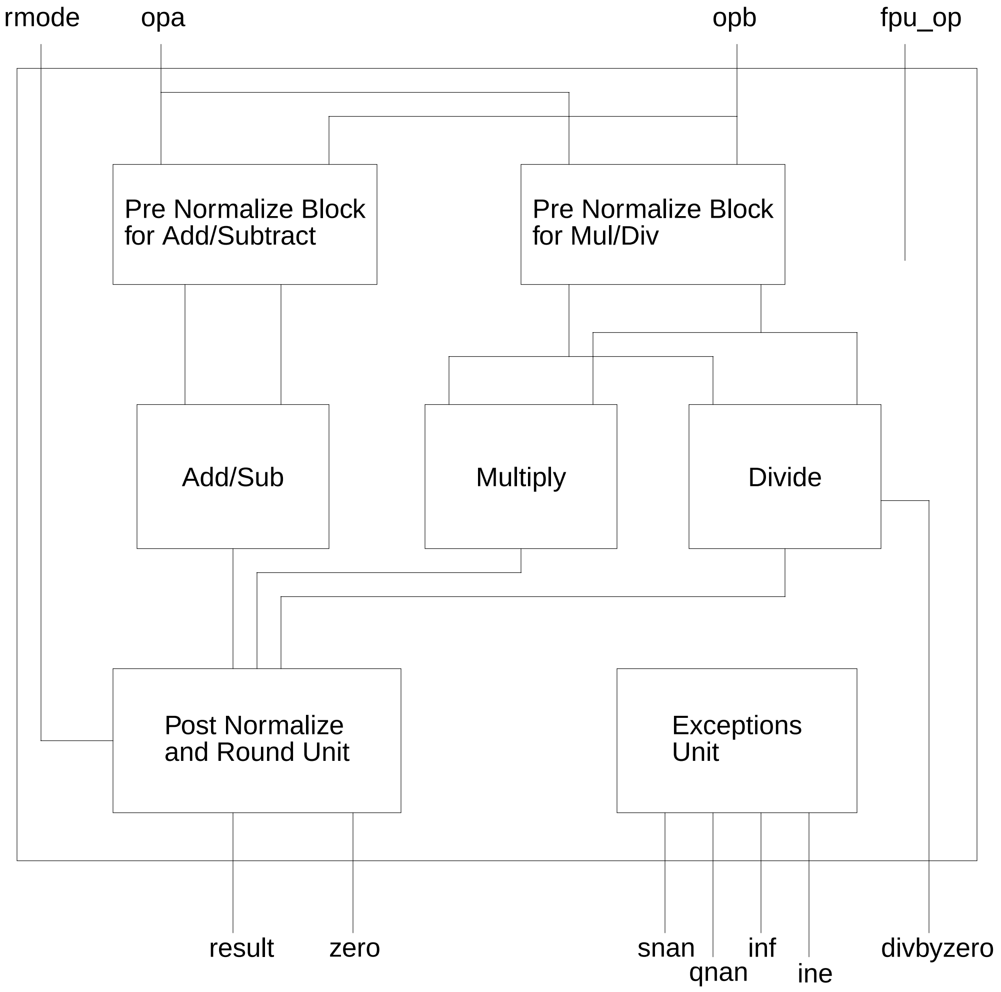
The QNAN output is asserted whenever the OUTUT of the FPU is NAN (always a quiet NAN).

When performing a floating point to integer conversion, the output (representing an integer) can take on forms of a NAN or INF, which are perfectly legal integers. In those cases the inf and nan outputs will not be asserted. However, if the input is a NAN, the proper nan output will be asserted.

2. FPU Architecture

This is a simple single precision floating point unit. Two pre normalization units adjust the fractions (mantissa) and exponents. One does it for add and subtract operation, the other for multiply and divide operations.
The Add/Sub, Mul, and Div, blocks, perform the actual addition subtraction, multiplication and division respectively.
A shared post normalization block, normalizes the fraction, the rounds it. The final result is packed in to a valid single precision floating point format result.

Below diagram illustrates the internal of this implementation (pipeline not shown).


rmode
opa
opb
fpu_op
Pre Normalize Block for Add/Subtract
Pre Normalize Block for Mul/Div
Add/Sub
Multiply
Divide
Post Normalize and Round Unit
Exceptions Unit
result
zero
snan
qnan
inf
ine
divbyzero

3. Interface

This table lists all inputs and outputs of the FPU and provides a general description of their functions.

Table 1:

Signal Name	With	Direction	Description
clk	1	Input	System Clock
rmode	2	Input	Rounding Mode
fpu_op	3	Input	Floating Point Operation Select
opa, opb	32	Input	Operand a and B
out	32	Output	Result Output
snan	1	Output	Asserted when either operand is a SNAN
qnan	1	Output	Asserted when output is a QNAN
inf	1	Output	Asserted when output is a INF
ine	1	Output	Asserted when the Result is Inexact
overflow	1	Output	Asserted when a overflow occurs
underflow	1	Output	Asserted when a Underflow occurs
div_by_zero	1	Output	Asserted when the fpu_op is set to divide and opb is zero
zero	1	Output	Asserted when the output is a numeric zero