



Block Diagram

FEATURES

- Fully Synthesizable RTL - Verilog
- IEEE 754R compliant (except underflow)
- Flag outputs support conditional branching or conditional execution
- Supports all IEEE rounding modes
- Supports all IEEE Exception flags
- Half Precision instructions
- Single Stage Pipeline
- Control / Status Registers
- Masked / Unmasked Exception control

OVERVIEW

The A2FH is a co-processor unit providing floating-point computation compliant with the *ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic* (IEEE-754R Standard). It is designed to provide a powerful floating-point functionality for low-power, low frequency applications.

The A2FH supports half precision operations in a 1- stage execution pipeline. The pipeline insures maximum performance in low-frequency applications, providing up to 200 MFLOPS on a 0.13u ASIC process. The host interface is clean and versatile, simplifying the interfacing to host processor pipelines.

IEEE-754R Compliance

The A2FH is designed to provide a powerful floating-point capability while minimizing die size cost. To minimize unnecessary design size, some of the rarely used features of the IEEE specification are not implemented directly in the hardware design. The following IEEE-defined operations are not directly supported in A2FH hardware, but can be supported with software support:

- Gradual Underflow
- Denormal Numbers

In place of gradual underflow, the A2FH implements a flush-to-zero approach when underflow occurs. This feature allows the A2FH to maintain a one-cycle throughput in all operand cases, and minimizes design size.

Optional Divide Unit

The divide unit within the A2FH design provides divide, square root, and remainder functions. In order to further minimize the design size, the A2FH can be synthesized with or without a divide unit. Many multi-media applications can be implemented without the use of divide functions. For customers who need the absolute minimum area, this option is a must.

Performance

without Divide unit:	10,000 NAND Gates
with Divide unit:	12,000 NAND Gates

Timing: 150 MHz clock on 0.180nm and 200 MHz on 0.130nm technology has been achieved

NOTE: The above performance data are estimates only, based on sample implementations using worst-case conditions. Achieved performance is highly dependent on the process technology, cell library, and synthesis tools used.

Instruction Timing

INSTRUCTION	Throughput/Latency
Add, Subtract, Difference, Multiply, Compare, Round-to-Integer	1
Single/Double Format Conversions, Integer Conversions	1
Min, Max, Clip	1
Absolute Value, Negate, Move	1
Divide	1 to 7
Square Root	1 to 6
Remainder, Modulus	1 to $(e/2 + 2)$

Notes:

- 1) Divide, Square Root, Remainder, and Modulus are implemented with an “early out” algorithm, where the iterative calculations are stopped if the current remainder becomes zero.
- 2) e = operand exponent difference