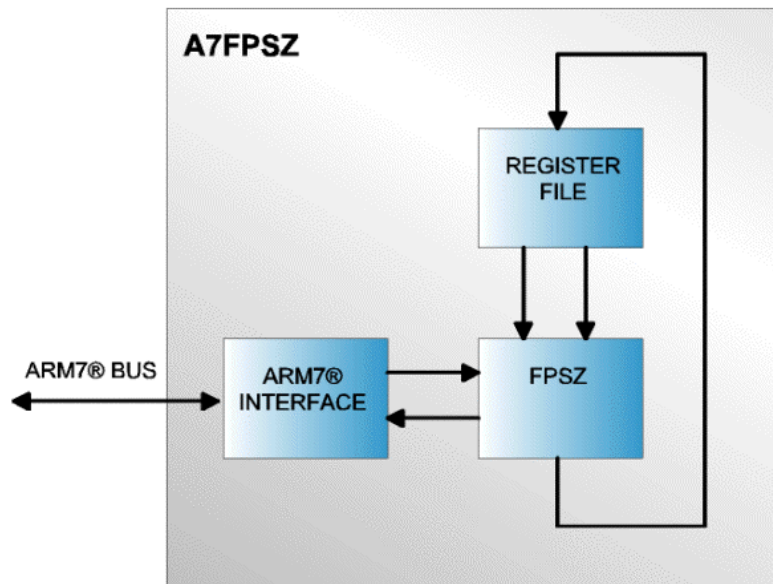


# A7FPSZ

## ARM7® IEEE-754 Floating Point Co-Processor Single Precision with Flush-to-Zero Underflow



### Overview

The A7FPSZ is a ARM7® co-processor unit providing floating-point computation compliant with the *ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic* (IEEE-754 Standard). It is designed to provide a powerful floating-point functionality for low-power, low frequency applications.

The A7FPSZ supports single precision operations that execute in 1 clock cycle. The design insures maximum performance in low-frequency applications, providing up to 133 MFLOPS on a 0.13u ASIC process. Via the co-processor interface, the A7FPSZ seamlessly integrates floating-point functionality into the ARM7 instruction set architecture.

### Features

- ARM7® Co-processor Interface
- Configurable Register File (4, 8, or 16 registers)
- IEEE-754 compliance (except underflow)
- Flush-to-Zero underflow implementation
- Single precision instructions
- 1-cycle execution for most instructions
- Instructions provided
  - Add/Subtract
  - Multiply
  - Divide (optional)
  - Remainder/Modulus (optional)
  - Square root (optional)
  - Floating Point Compare
  - Floating Point ↔ Integer Conversions
  - Round to Integer
  - Absolute Value/ Negate
  - Move
- Flag outputs support conditional branching or conditional execution
- All IEEE rounding mode supported
- All IEEE exception flags supported
- Masked and unmasked exception control
- Control and status register

## IEEE-754 Compliance

The A7FPSZ 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 FPSZ hardware, but can be supported with software support:

- Gradual Underflow
- Denormal Numbers

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

## Optional Divide Unit

The divide unit within the A7FPSZ design provides divide, square root, and remainder functions. In order to further minimize the design size, the A7FPSZ 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

Size: 40,000 NAND Gates, w/o divide unit  
48,000 NAND Gates, w divide unit

Timing: 100 MHz on 0.18u  
133 MHz on 0.13u

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, Multiply, Compare, Round to Integer	1
Single/Double Format Conversions Integer Conversions	1
Absolute Value, Negate, Move	1
Divide	1 to 14
Square Root	1 to 13
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

Technical data is subject to change without notice. All trademarks are registered trademarks of their respective owners.  
Copyright © GB3 Digital Systems 2005, All rights reserved.