MIPS Floating Point Architecture

- * Hardware + software fully conforms to IEEE 754 and also fully supports the standard's recommendations
- * All floating point operations are fully emulated in kernel
 - * Programs run correctly in systems without FP hardware
 - * Particular hardware implementations may trap to software for any "difficult" operation
- * Exceptions are recorded in sticky and non-sticky flags, and trap if enabled
- * Exceptions are precise

EPC identifies faulting instruction; it and all subsequent instructions have not been executed; all previous instructions have been executed

MIPS Floating Point Hardware

- * IEEE 754 Single and Double formats supported in R3010 FP coprocessor
 - * R3010 traps to kernel software for denorms and NaNs
 - * Low-latency operations:

dp operation	cycles	ns
add, subtract	2	80
multiply	5	200
divide	19	760
move/abs/neg/compare	1	40

^{*} Integer, load/store, fp add, fp multiply, and fp divide units can operate in parallel

^{* 4} DP, 7 SP LINPACK MFLOPS

MIPS Math Library

- * Single and double versions of most functions
- * 0.5 ulp SQRT done in software using Kahan algorithm
- * LOG, EXP, SIN, COS, TAN, SINH, COSH, TANH based on Cody and Waite algorithms
 - * rational or polynomial approximations
 - * generally 100 to 130 cycles with 1.5ulp accuracy
- * Other functions based on 4.3bsd math library (including DREM, POW, LOG1P, EXPM1, ATAN, ...)
- * Binary <-> decimal conversion done using 64-bit integer arithmetic and fully accurate powers of 10
- * Math library functions do not act as atomic operations; there is only partial attempt to obey rounding mode and signal exceptions as in hardware operations
- * Math library does return NaN, Infinity etc. as appropriate sqrt(-1) = NaN log(0) = -Infinity log(-1) = NaN exp(1000) = Infinity sin(1e38) = NaN etc.

MIPS Exceptions

- * Exception traps disabled by default
- * Enabled exceptions generate SIGFPE signal
 - * Handler can examine instructions and operands and continue with its own value
 - * Library includes SIGFPE handler for counting exceptions and kernel emulations, and reporting locations of each
- * get_fpc_csr, set_fpc_csr read control status register, allowing programs to examine and reset modes, flags, and enables
- * swapRM, swapINX also supported