The Difference Between x87 Instructions FSIN, FCOS, FSINCOS, and FPTAN and Mathematical Functions sin, cos, sincos, and tan

Warren Ferguson, Marius Cornea, Cristina Anderson, Eric Schneider

Introduction

In the early 1980s, the Intel® 8087 Math Coprocessor introduced hardware support for a small set of elementary transcendental functions (trigonometric, inverse trigonometric, exponential, and logarithmic), accessible through x87 instructions. In the 1990s Intel replaced the 8087's CORDIC-based approximations of the elementary transcendental functions with polynomial-based approximations. These newer polynomial-based approximations provide a large degree of backwards compatibility with the CORDIC based approximations by approximating precisely the same functions, but with greater overall accuracy and speed.

The purpose of this paper is to inform users of many of the implementation details of the x87 trigonometric instructions. If users find that the x87 elementary transcendental instructions are not adequate for their needs, they should consider using the software libraries of transcendental functions available through the Intel® C++ Compiler and Intel® Fortran Compiler, the Intel® Math Kernel Library (Intel® MKL) product, or libraries from other providers. Just as double precision arithmetic is provided to users when single precision floating-point arithmetic is not adequate, so these software library functions provide more accurate approximations when the x87 instructions are not adequate.

Rounded Pi

The value of the mathematical constant $\pi \cong 3.14159265358979 \ldots$ is (in hexadecimal format):

$$\pi \stackrel{\text{def}}{=} (0.\underbrace{\text{C90FDAA22168C234C}}_{68 \text{ bits}} \text{4C6628B80DC1CD12902} \ldots)_{16} \cdot 2^2$$

The value p that the x87 uses as its approximation of π is the value of π rounded-to-nearest to its leading 68 bits (of which the last two bits are zeros):

$$p \stackrel{\text{def}}{=} (0.\text{C90FDAA22168C234C})_{16} \cdot 2^2 .$$

Note that p differs from π by a little more than one unit in the 70th bit of π.

The bit-length of p, measured as the number of bits between its most significant and least significant bit inclusively, is 66. (This is why Volume 1 of the Intel® Software Developer's Manuals refers to p as a 66-bit approximation of π.)

Observe that

$$\pi - p = (0.4C6628B80DC1CD12902\ldots)_{16} \cdot 2^{-66} = (0.98CC51701B839A25204\ldots)_{16} \cdot 2^{-67}$$

and, as a result, the relative error e in the approximation of p by π is

$$e \stackrel{\text{def}}{=} \frac{\pi-p}{p} = \frac{\pi}{p} - 1 \cong (1.28741467897512077\ldots)_{10} \cdot 10^{-21} \cong 1.5 \cdot 2^{-70}.$$

Definition of the x87 Trigonometric Approximations

FSIN, FCOS, FSINCOS, and FPTAN are x87 double-extended precision instructions that compute approximations of the following functions:

$$FSIN(x) \cong \sin(x \cdot \pi/p)$$

$$FCOS(x) \cong \cos(x \cdot \pi/p)$$

$$FSINCOS(x) \cong \{\sin(x \cdot \pi/p), \cos(x \cdot \pi/p)\}$$

$$FPTAN(x) \cong \{\sin(x \cdot \pi/p)/\cos(x \cdot \pi/p), 1\}$$

for all x for which $|x| < 2^{63}$. For any one of the trigonometric functions on the RHS, the error in the corresponding x87 approximation on the LHS is less than 1 ulp in round-to-even mode, and 1.5 ulps[1] in the other rounding modes.

Because the functions FSIN, FCOS, FSINCOS, and FPTAN share $2p$ as a common period, and $2p$ is the value of 2π rounded-nearest to its leading 68 bits, they are said to be rounded-period approximations.

Derivation of the Value of the x87 Trigonometric Approximations

The value of these rounded-period trigonometric functions at a double-extended precision argument x is obtained through a three-step process:

1. Reduction: Compute $\{N, r\}$ so $x = N \cdot \left(\frac{p}{2}\right) + r$ exactly; N is an integer and $-p/4 < r < p/4$.
2. Approximation: Compute double-extended precision approximations of $\{\sin(x \cdot \pi/p), \cos(x \cdot \pi/p)\}$.
3. Reconstruction: The two least significant bits of N allow the recovery of the desired rounded-period trigonometric function.

The reduction step is performed, essentially, through the use of a floating-point remainder instruction. This remainder instruction is exact, and so the value of r is exact. We know that r can never attain either of the values $\pm p/4$ because, if it did, then it would follow that $x = (2N \pm 1) \cdot p/4$; however this is impossible, because the RHS has a significand that is at least 66 bits long, and we know that the double-extended precision x has a significand that is at most 64 bits long. A similar argument tells us that r can

---

[1] See "On the definition of ulp(x)" by Jean Michel Muller. Roughly speaking, for any real number x and a given floating-point format, $ulp(x)$ is the distance between the two floating-point numbers in that format that are nearest to x.

be zero if and only if x is zero. (Backwards compatibility with the original 8087 requires that this reduction step remain unchanged.)

Now $\sin(r \cdot \pi/p)$ is an odd function on the interval $-p/4 < r < p/4$, so it is approximated by polynomials that are a sum of constants multiplied by odd powers of r. Similarly, $\cos(r \cdot \pi/p)$ is an even function on the same interval, so it is approximated by polynomials that are a sum of constants multiplied by even powers of r. As a result, FSIN and FPTAN are odd functions of their argument, while FCOS is an even function of its argument, i.e.:

$$FSIN(-x) = -FSIN(x)$$

$$FCOS(-x) = FCOS(x)$$

$$FPTAN(-x) = -FPTAN(x) .$$

The reconstruction step is based on the addition rules for sine and cosine. These rules tell us that

$$\sin\left(x \cdot \frac{\pi}{p}\right) = \sin\left(N \cdot \frac{\pi}{2} + r \cdot \frac{\pi}{p}\right) = \sin\left(N \cdot \frac{\pi}{2}\right) \cdot \cos\left(r \cdot \frac{\pi}{p}\right) + \cos\left(N \cdot \frac{\pi}{2}\right) \cdot \sin\left(r \cdot \frac{\pi}{p}\right)$$

$$\cos\left(x \cdot \frac{\pi}{p}\right) = \cos\left(N \cdot \frac{\pi}{2} + r \cdot \frac{\pi}{p}\right) = \cos\left(N \cdot \frac{\pi}{2}\right) \cdot \cos\left(r \cdot \frac{\pi}{p}\right) - \sin\left(N \cdot \frac{\pi}{2}\right) \cdot \sin\left(r \cdot \frac{\pi}{p}\right)$$

and leads to the following table that describes how the rounded-period sine and cosine at argument x is related to the rounded-period sine and cosine at argument r. As this table shows, the rounded-period sine or cosine at argument x is, to within a sign, its value at the reduced argument r.

Function	$N \bmod 4 = 0$	$N \bmod 4 = 1$	$N \bmod 4 = 2$	$N \bmod 4 = 3$
$sin\left(x \cdot \dfrac{\pi}{p}\right)$	$\sin\left(r \cdot \dfrac{\pi}{p}\right)$	$\cos\left(r \cdot \dfrac{\pi}{p}\right)$	$-\sin\left(r \cdot \dfrac{\pi}{p}\right)$	$-\cos\left(r \cdot \dfrac{\pi}{p}\right)$
$cos\left(x \cdot \dfrac{\pi}{p}\right)$	$\cos\left(r \cdot \dfrac{\pi}{p}\right)$	$-\sin\left(r \cdot \dfrac{\pi}{p}\right)$	$-\cos\left(r \cdot \dfrac{\pi}{p}\right)$	$\sin\left(r \cdot \dfrac{\pi}{p}\right)$

Further Properties of the x87 Trigonometric Approximations

Because the rounded-period approximations of the sine and cosine are evaluated accurately at the same argument $x \cdot \pi/p$, we expect that any mathematical identity between them also holds, with small absolute error, when their values are replaced by the values of their corresponding rounded-period approximations. For example, because

$$\sin(x)^2 + \cos(x)^2 = 1$$

then we expect that

$$FSIN(x)^2 + FCOS(x)^2 \cong 1$$

with small absolute error; here small means a value with a magnitude that is a modest multiple of 2^{-63}.

To relate the sine and cosine at x to the sine and cosine at $x \cdot \pi/p$ note that

$$x = x \cdot \frac{\pi}{p} - x \cdot \left(\frac{\pi}{p} - 1\right) = x \cdot \frac{\pi}{p} - x \cdot e \ .$$

where e is as defined at the end of the "Rounded Pi" section.

The addition rules for sine and cosine tell us that

$$\sin(x) = \sin\left(x \cdot \frac{\pi}{p} - x \cdot e\right) = \sin\left(x \cdot \frac{\pi}{p}\right) \cdot \cos(x \cdot e) - \cos\left(x \cdot \frac{\pi}{p}\right) \cdot \sin(x \cdot e)$$

$$\cos(x) = \cos\left(x \cdot \frac{\pi}{p} - x \cdot e\right) = \cos\left(x \cdot \frac{\pi}{p}\right) \cdot \cos(x \cdot e) + \sin\left(x \cdot \frac{\pi}{p}\right) \cdot \sin(x \cdot e)$$

Because $\{FSIN(x), FCOS(x)\} \cong \{\sin(x \cdot \pi/p), \cos(x \cdot \pi/p)\}$ with an error of less than 1.5 double-extended precision ulps, we would expect

$$\sin(x) \cong FSIN(x) \cdot \cos(x \cdot e) - FCOS(x) \cdot \sin(x \cdot e)$$

$$\cos(x) \cong FCOS(x) \cdot \cos(x \cdot e) + FSIN(x) \cdot \sin(x \cdot e)$$

to hold with small absolute error. Recall that $|x| < 2^{63}$ and $e \cong 1.5 \cdot 2^{-70}$, so $|x \cdot e| \lesssim 1.5 \cdot 2^{-7} \cong$ 0.012 radians $\cong 0.68$ degrees. Therefore, $\cos(x \cdot e) \cong 1 + \mathcal{O}((x \cdot e)^2)$ and $\sin(x \cdot e) \cong (x \cdot e) + \mathcal{O}((x \cdot e)^3)$. From these estimates, we conclude that

$$\sin(x) \cong FSIN(x) - (x \cdot e) \cdot FCOS(x) + \mathcal{O}((x \cdot e)^2)$$

$$\cos(x) \cong FCOS(x) + (x \cdot e) \cdot FSIN(x) + \mathcal{O}((x \cdot e)^2)$$

and so

$$|\sin(x) - FSIN(x)| \leq |x \cdot e| + \mathcal{O}((x \cdot e)^2)$$

$$|\cos(x) - FCOS(x)| \leq |x \cdot e| + \mathcal{O}((x \cdot e)^2).$$

Another way to think about the quality of these approximations is based on the fact that the value of a rounded-period trigonometric function at argument x is an accurate approximation of the corresponding trigonometric function at an argument $x' = x \cdot \pi/p$ that lies very near x; the error in approximating x by x' amounts to something much less than 1 double-extended precision ulp in x. As long as the value of the expression being computed is insensitive to such small changes in the argument x, it is safe to approximate the trigonometric functions in that expression by their rounded-period counterparts. Here we are appealing to the technique of backward error analysis, whose goal is to recast the approximate evaluation of an expression of exact arguments as an exact evaluation of the same expression but using approximations of the arguments.

For example, the distance from the origin to the point $\{7 \cdot FCOS(3x), -5 \cdot FSIN(3x)\}$ is an accurate estimate of the distance from the origin to the point $\{7 \cdot \cos(3x), -5 \cdot \sin(3x)\}$; this distance calculation is insensitive to changes in the argument of the trigonometric functions.

On the other hand, it would not be safe to use $FSIN(x)$ to approximate $\sin(x)$ for arguments x near a nonzero multiple of π. When $x \stackrel{\text{def}}{=} n \cdot \pi$ with $n \neq 0$, then for small $|\delta|$ we know that

$$\sin(x + \delta) = \cos(n \cdot \pi) \sin(\delta) = (-1)^n \sin(\delta) \cong (-1)^n \delta,$$

and so a change of δ in x causes a change of similar magnitude in the value of the sine. However, for $n = 0$ it is safe to approximate $\sin(x)$ with $FSIN(x)$; each rounded-period trigonometric function is a very accurate approximation of the corresponding trigonometric function as the magnitude of the argument tends to zero (for such small magnitude arguments x, the reduction step yields $\{N, r\} = \{0, x\}$).

In a similar manner, *FCOS(x)* should not be relied on as an accurate approximation of cos(x) near odd multiples of π/2, and *FPTAN(x)* should not be relied on as an accurate approximation of $\tan(x)$ near multiples of π/2, odd or even.

For example, a graphical representation of the ulp error when $FSIN(x)$ is used to approximate the mathematical function $\sin(x)$ is shown in the following graph:
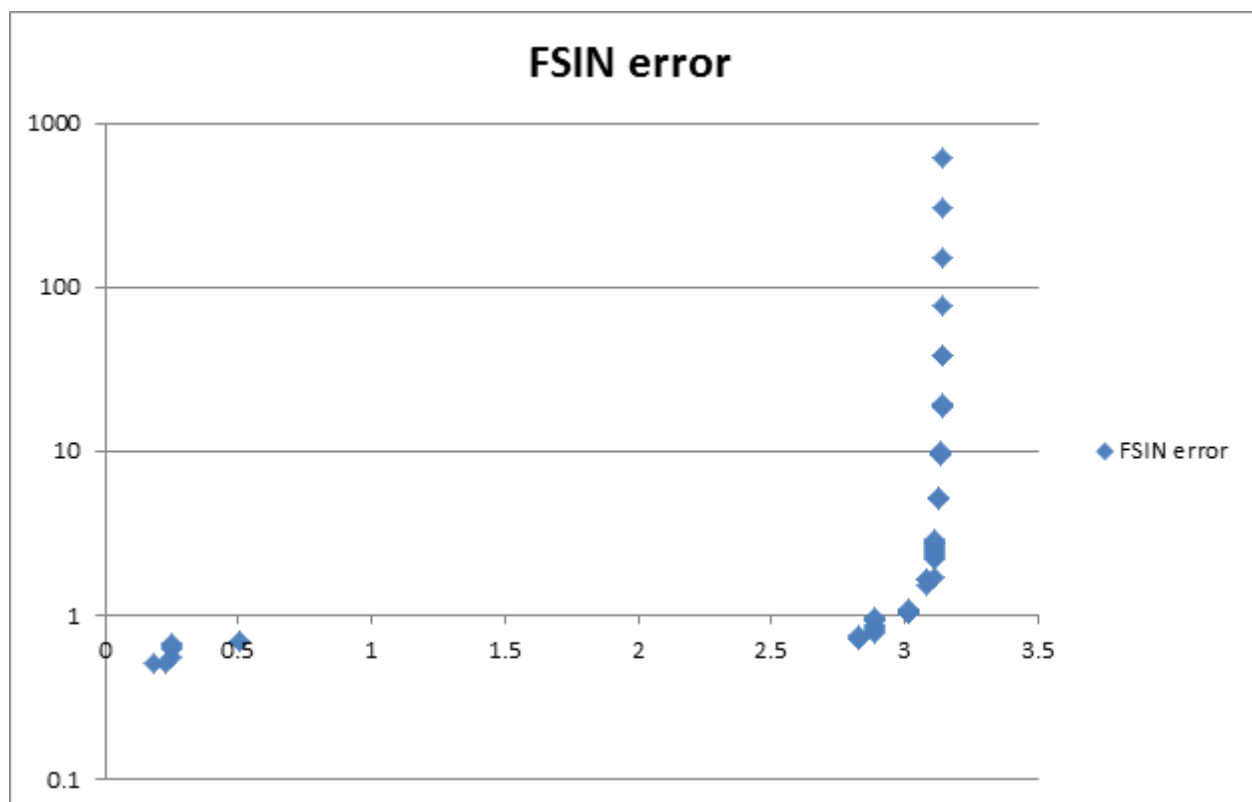
The graph shows how the ulp error increases sharply as the argument x of FSIN approaches the value π.

Finally, consider the problem of computing the value of an expression that involves the values of the trigonometric functions at x, where x is the result of a previous computation and it contains itself an error of several double-extended precision ulps.

If the value of that expression is insensitive to changes in x, then the value of that expression determined via the rounded-period trigonometric functions in place of the actual trigonometric functions causes an additional error no larger than that already present in the approximate value of x. The following table displays the radian and degree measures of a 1 double-extended precision ulp

change in $x \stackrel{\text{def}}{=} (1 + f) \cdot 2^e$. Here the double-extended precision x has a fractional portion f and exponent e, so a 1 ulp change in x amounts to 2^{e-63}.

e	1 ulp change in x	
	radians	degrees
-3	1.36E-20	7.77E-19
2	4.34E-19	2.48E-17
7	1.39E-17	7.95E-16
12	4.44E-16	2.54E-14
17	1.42E-14	8.14E-13
22	4.55E-13	2.61E-11
27	1.46E-11	8.34E-10
32	4.66E-10	2.67E-08
37	1.49E-08	8.54E-07
42	4.77E-07	2.73E-05
47	1.53E-05	8.74E-04
52	4.88E-04	2.80E-02
57	1.56E-02	8.95E-01
62	5.00E-01	2.86E+01

Conclusion

It is important to remember that $FSIN(x)$, $FCOS(x)$, $FSINCOS(x)$, and $FPTAN(x)$ are approximations of the corresponding mathematical functions of argument $(x \cdot \pi/p)$, and not of argument (x). Therefore they can only be used to approximate the mathematical functions *sin(x)*, *cos(x)*, and *tan(x)* on a limited domain, outside of which a more accurate argument reduction technique is necessary in order to contain the errors, or, alternatively, accurate software implementations of these mathematical functions may to be used.

The limitations of using a 68-bit approximation of π are also explained in the *Intel® 64 and IA-32 Architectures Software Developer Manuals*, available online at http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html. See Vol. 1, section 8.3.8 "Approximation of Pi", and the instruction manual pages in Vol. 2A, section 3.2.