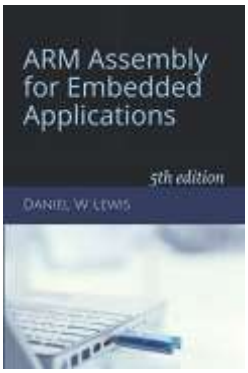


Programming Lab 9F

Inverse Square Root

Topics: Floating-point arithmetic

Click to download Lab9F-Main.c

Prerequisite Reading: Chapters 1-9
Revised: June 22, 2021

Background[1]: The inverse square root ($1/\sqrt{x}$) is used extensively in computer graphics programs to compute angles of incidence and reflection for lighting and shading. Normally, this would require using the two slowest floating-point instructions – `VSQRT` and `VDIV`. However, in 1999 a faster alternative based on Newton's root-finding method was used in the source code of the game *Quake III Arena*.

The algorithm treats the bits representing the floating-point number as a 32-bit integer, logically shifts them right by one bit, and then subtracts the result from the number 0x5F3759DF, which is a floating-point representation of an approximation of $\sqrt{2^{127}}$. This provides a very good first approximation of the inverse square root of the input. Treating the bits of this result as a floating-point number, it then runs one iteration of Newton's method, yielding a more precise final approximation.

Assignment: The main program will compile and run without writing any assembly. However, your task is to create equivalent replacements in assembly language for the two C functions shown below and found in the main program. The original C versions of the functions have been defined as "weak" so that the linker will automatically replace them in the executable image by those you create in assembly; you do not need to remove the C versions.

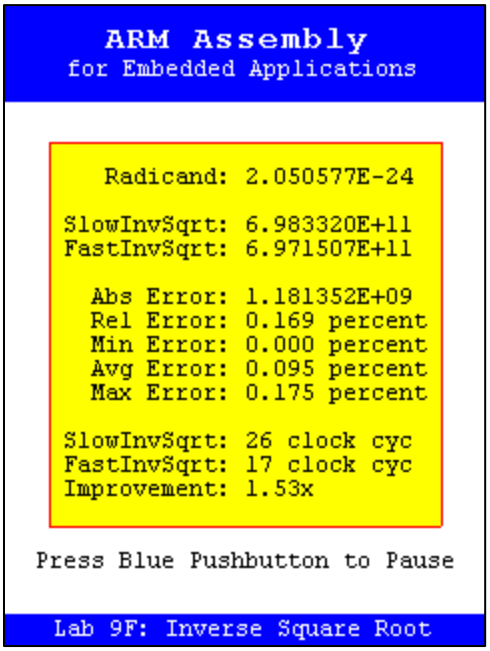
```
float SlowInvSqrt(float radicand) ;
```

Uses square root and divide instructions to compute the result.

```
float FastInvSqrt(float radicand) ;
```

Newton's method with first approximation described above.

Test your implementation using the C main program. If your code is correct, the program will call the two functions continuously with random floating-point values, displaying their resulting values, the absolute, relative, minimum, average and maximum error, and the execution time performance. Pressing and holding the blue pushbutton will pause the execution; releasing the pushbutton allows execution to resume. If an error occurs, an error message will be displayed as **white text on a red background** and execution halted.



[1] https://en.wikipedia.org/wiki/Fast_inverse_square_root