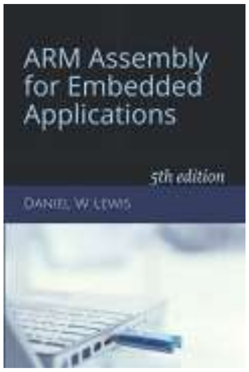


Programming Lab 11E

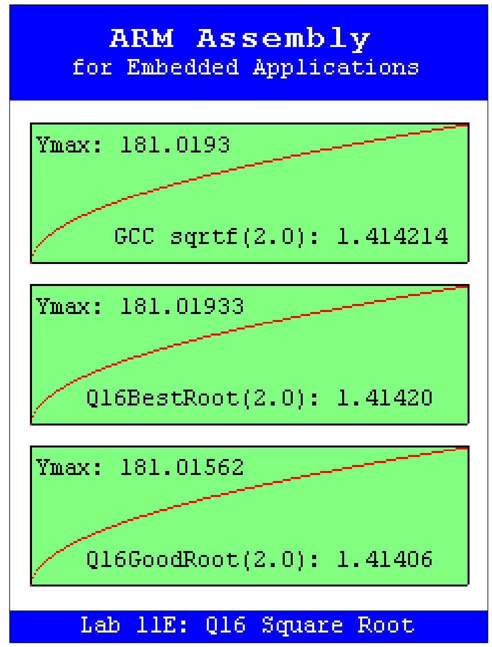
Q16 Square Root

Click to download Lab11E-Main.c

Topics: Q16 fixed-point arithmetic, square root by abacus algorithm.

Prerequisite Reading: Chapters 1-11

Revised: June 6, 2021



Background: A real number X is represented in Q16 format by an integer that is the product of X and 2^{16}. Since the representation is actually an integer, a simple modification of an integer square root algorithm may be used to compute the Q16 square root:

Let X and R be real numbers, where:

$$R = \sqrt{X} \tag{1}$$

In Q16 format, X and R are represented by integers N_x and N_R, where:

$$X = N_X/2^{16} \text{ and } R = N_R/2^{16} \tag{2,3}$$

Substituting (2) and (3) into (1) yields:

$$N_R/2^{16} = \sqrt{N_X/2^{16}} \tag{4}$$

Multiplying both sides by 2^{16} gives:

$$N_R = 2^{16}\sqrt{N_X/2^{16}} = 2^8\sqrt{N_X} \tag{5}$$

I.e., a good approximation to the square root of a Q16 real number X may be computed by shifting the integer square root of N_X left by 8 bits. However, this forces the least-significant 8 bits of the result to zeroes, thus limiting the precision to the remaining 8 fractional bits. A better approximation with a full 16 bits of precision may be computed by multiplying *before* taking the integer square root:

$$N_R = 2^8\sqrt{N_X} = \sqrt{2^{16}N_x} \tag{6}$$

Assignment: The main program includes two C functions that are modified versions of a "digit-by-digit" algorithm for computing the integer square root that was originally developed for use with an abacus[1]:

```
typedef int32_t Q16 ;

Q16 Q16GoodRoot(Q16 radicand) ;     // Precision: about ±.001   (implementation required)
Q16 Q16BestRoot(Q16 radicand) ;     // Precision: about ±.00001 (optional for extra credit)
```

Note: The main program may be compiled and executed without writing any assembly. However, your task is to create faster assembly versions of these two functions using the C versions to guide your implementation. The original C functions have been defined as "weak", so that the linker will automatically replace them in the executable image by those you create in assembly; there is no need to remove the C versions. Both C functions call an intrinsic function named `__CLZ`; your assembly language version may replace each call by a `CLZ` instruction.

Suggestion: Code and test your replacement functions one at a time in the order shown above. If your code is correct, the display should look like the image shown. The program displays square roots computed by the two Q16 functions as well as that computed by the single precision `sqrtf` function from gcc's floating-point library. The horizontal axis of each graph covers the entire range of non-negative Q16 values, while the vertical axis is the square root. For comparison, each graph also shows the numeric values computed for the square root of 2 and the square root of the maximum Q16 value.

---

[1] http://medialab.freaknet.org/martin/src/sqrt/sqrt.c