


Programming Lab 11B

Solving Quadratics with Q16 Fixed-Point Reals

Topics: Representation of real numbers using Q16 fixed-point.

Prerequisite Reading: Chapters 1-11

Revised: May 27, 2021

 Click to download
Lab11B-Main.c

Background: This assignment is identical to Lab 9A except that instead of hardware floating-point instructions, your code will use Q16 fixed-point reals. Q16 uses 32-bit integers to hold the bit patterns of the fixed-point real numbers:

```
typedef int32_t Q16 ;
```

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 following four functions found in the C main program. The original C versions 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. This allows you to create and test your assembly language functions one at a time.

Q16 Root1(Q16 a, Q16 b, Q16 c) ;

Computes the root given by $\frac{-b + \sqrt{\text{Discriminant}(a,b,c)}}{2a}$

Q16 Root2(Q16 a, Q16 b, Q16 c) ;

Computes the root given by $\frac{-b - \sqrt{\text{Discriminant}(a,b,c)}}{2a}$

Q16 Quadratic(Q16 x, Q16 a, Q16 b, Q16 c) ;

Computes the quadratic, $ax^2 + bx + c$
Most efficient order of evaluation: $c + x(b + ax)$

Q16 Discriminant(Q16 a, Q16 b, Q16 c) ;

Computes the value of the discriminant, $b^2 - 4ac$
Functions Root1 and Root2 should call this function.
Most accurate order of evaluation: $b^2 - (4a)c$ -or- $b^2 - a(4c)$

Each multiplication should be coded using an SMULL followed by a shift.

To simplify your task, your assembly language functions should make use of the following two C functions defined in the main program. Your assembly language code will likely need to push and pop some registers; be sure that the total number of registers you push and pop is even so that the address held in the stack pointer remains a multiple of eight to satisfy the [data alignment requirements](#) of compiled C code.

```
Q16 Q16Quotient(Q16 a, Q16 b) ;           // returns a / b  
Q16 Q16SquareRoot(Q16 x) ;               // returns sqrt(x)
```

Test your code with the main program. If your code is correct, the display should look similar to the image shown, the sliders can be used to vary the coefficient values, and pressing the blue pushbutton will restore the initial conditions. Otherwise, incorrect return values will cause an error message to be displayed as **white text on a red background** and the program will be halted.

