IAR Application Note G-002

Fast square root in C

SUMMARY

This application note describes a solution for a fast square root routine in C. Both the parameter and the result are 16-bit unsigned ints. Even though it is not 100% precise, the result can be used with many applications in the real world running in real time.

KEYWORDS

sqrt, algorithm

The Problem to be Solved

Designers often face the square root problem, especially when doing FFT, or getting the RMS value of signals. Using this piece of software, the designer can save a lot of design and runtime.
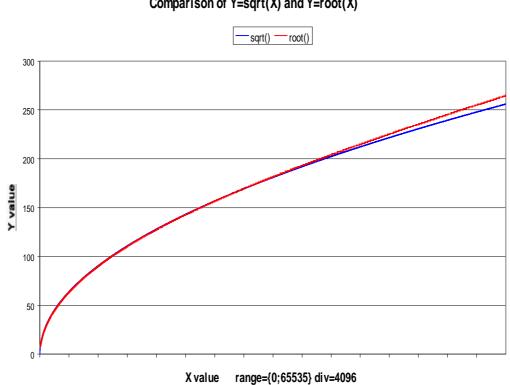
The Solution

The solution is best explained by looking at the program routine itself. The program was tested and executed on a PC running IAR Embedded Workbench™ and IAR C-SPY™ Debugger. The comparison results are shown in the figure below.

```
/***********************************************************/
#include <math.h>                          /* only for sqrt()  */
#include <stdio.h>                          /* only for printf() */
unsigned int iii;                           /* test variable    */
/***********************************************************/
unsigned int root(unsigned int x){
unsigned int a,b;
 b     = x;
 a = x = 0x3f;
 x     = b/x;
 a = x = (x+a)>>1;
 x     = b/x;
 a = x = (x+a)>>1;
 x     = b/x;
 x     = (x+a)>>1;
 return(x);                          }
/***********************************************************/
void main(void){          /* creates text data file for Excel */

  for(iii=0;iii<(0xfffc);iii=iii+4){
      printf("\n %3.1f; %u ",(float)sqrt(iii), root(iii));
  }
}
/***********************************************************/
```

The User Benefits

Despite a very short runtime for getting the square root results, we do not need a fast look-up table that requires a lot of data memory space.

Comparison of Y=sqrt(X) and Y=root(X)

sqrt() — root()

Y value

X value   range={0;65535} div=4096