RL78/G10

Square Root Program CC-RL

R01AN3079EJ0100
Rev. 1.00
Feb. 03, 2016

Introduction

This application note explains the RL8/G10 program for finding the square root of a 32-bit binary number. The interface is compatible with C language, BC and AX registers are used for input parameters, and the BC register is used for output parameters.

Target Device

RL78/G10

This application note was developed for use with all members of the RL78/G series. However, when using the application for a microcomputer (MCU) other than RL78/G10, please evaluate thoroughly based on your target MCU's specifications.

Contents

1. Specifications

This application note describes a program for finding the square root of a 32-bit binary number using as little memory as possible. The interface is compatible with C language, BC and AX registers are used for input parameters, and the BC register is used for output parameters. The application can easily be called up by a C language program.

- Table 1.1 lists the arithmetic processing used in the application.

Table 1.1 Arithmetic Processing Used in Application

Function	Operation description
Square root operation	Inputs a 32-bit binary number and produces its square root as a 16-bit binary number.

2. Operating Conditions

The sample code in this application note runs under the following operating conditions.

Table 2.1 Operating Conditions

Item	Description/Specification
MCU used	RL78/G10 (R5F10Y16)
Operating frequency	● High-speed on-chip oscillator clock (HOCO): 20MHz ● CPU/peripheral hardware clock: 20MHz
Operating voltage	5.0 V (can run on a voltage range of 2.9 V to 5.5 V.) SPOR detection voltage: Rising edge voltage: 2.90V : Falling edge voltage: 2.84V
Integrated development environment (CS+)	CS+ for CC V3.01.00 from Renesas Electronics Corp.
Assembler (CS+)	CC-RL V1.01.00 from Renesas Electronics Corp.
Integrated development environment (e^2 studio)	e^2 studio V4.0.2.008 from Renesas Electronics Corp.
Assembler (e^2 studio)	CC-RL V1.01.00 from Renesas Electronics Corp.
Usage environment	RL78/G10 simulator

3. Software Explanation

3.1 Operation Outline

This square root function uses registers BC and AX as the argument and returns the result through the BC register. To make the program as small as possible, most processing is performed using registers without regard to expandability. The sample program offered in this application note only comprises a sample routine for calculating the square root. Please note that this application does not include a main processing function.

This square root function uses registers BC and AX as the argument and returns the result through the BC register.

3.2 Square Root Operation Concept

There are various ways to find the square root using any computer. This application uses the extraction method. The algorithm is the same as that described in the H8/300H Tiny Series application note titled “Square Root of a 32-bit Binary Number (SQRT)” (RJJ06B0075).

(1) Registers and memories used in operations

Table 3.1 lists the registers and memories used in the application and Figure 3.1 shows the corresponding configuration.

Table 3.1 Registers and Memories

Register/Memory	Description
AX register	Lower-order word of input data and shift work area
BC register	Higher-order word of input data and shift work area
DE register	Work area
HL register	Result area
DIGITU1	Work area extension
DIGITU2	Calculation-result area extension

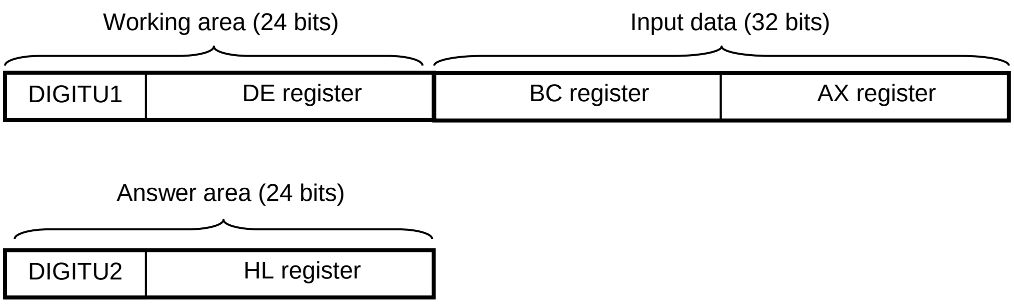


Figure 3.1 Register and Memory Configuration

The working area is cleared to 0, and then the input data for calculation is shifted from right to left in 2-bit units.

The calculation-result area stores the results of each step in the calculation and shifts 1 bit left for each calculation. When the LSB of the 24 bits is 1, the program subtracts the value in the calculation-result from the value in the working area. Although the final result is 16-bit number, the value of the calculation result is shifted 1 bit left, making it 17 bits. Because the calculation processing cannot be completed in just 16 bits, the 8-bit variable DIGITU2 has been added to higher-order bits of the HL register.

(2) Program Operations

The program performs the following operations:

① Initializes the following to 0: DE register and variable DIGITU1 for work, HL register, and variable DIGITU2 for calculation-result.

② Shifts the 32 bits of input data in 2-bit units to the DE register and variable DIGITU1 (for work).

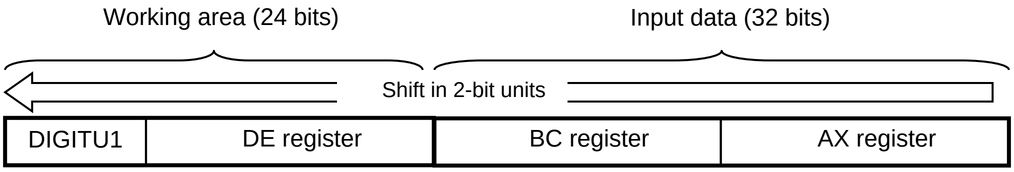


Figure 3.2 Shift In Input Data

③ Shifts the HL register and variable DIGITU2 (for calculation result) 1 bit left, and sets the LSB in the HL register to 1.

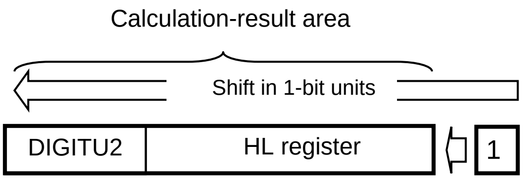


Figure 3.3 Preparation of Result Area

④ Subtracts the calculation-result area value from working area value.

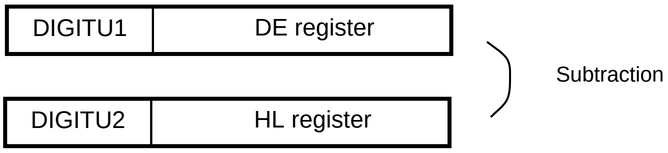


Figure 3.4 Subtract Calculation-Result Area from Working Area

⑤ If the value is subtracted from the working area, the calculation-result area (only L register) is incremented by 1. At this point, twice the value of the result is stored. When the value of the L register, originally xxxx xx01B, is incremented, it becomes xxxx xx10B and does not need to be carried to the higher-order.

⑥ If the value was not subtracted, the working area is returned to its original value and the L register LSB is set to 0. At this point, a value twice that of the result is stored in the calculation-result area.

⑦ Repeats steps 2 to 6 sixteen times.

⑧ When the calculation-results area is right shifted by 1 bit, the result is obtained and stored in the BC register.

This program does not execute any other processes. Therefore, the operation is executed 16 times for all input data.

3.3 Library Functions (subroutines)

Table 3.2 lists the library function used in this program.

Table 3.2 Library Function (subroutines)

Function Name (subroutine)	Description
__ssqrt	Square root operation for 32-bit data

3.4 Library Function (subroutine) Specifications

The following are the specifications of library functions (subroutines) used in the sample code.

Function Name: __ssqrt

Outline	Square root operation processing of 32-bit input data	
Description	Calculates square root of 32-bit input data, returns 16-bit result.	
Argument	BC register	Higher 16 bits of input data
	AX register	Lower 16 bits of input data
Return Value	BC register	Operation result:16-bit data
Notes	Stores same value in AX register and returns.	
	Uses 4 bytes to call stack, 2 bytes for internal processing, and 4 bytes to prevent destruction of register value.	

3.5 How to Use the Library Function (subroutine)

This program was made as a library. To embed as a library, add SQRTLIB.lib to your project. To embed as part of your application, add SQRTLIB.asm to your project.

The program is set with SSQRT for Assembler and __ssqrt for C language. When using the program from Assembler, make sure you declare the following:

```
.EXTRN  SSQRT                    ; 32-bit square root
```

Next, set the upper 16 bits of the BC register and the lower 16 bits of the AX register, and call SSQRT in a subroutine. The result of the square root-extraction will be set in the BC register (and the AX register) and returned.

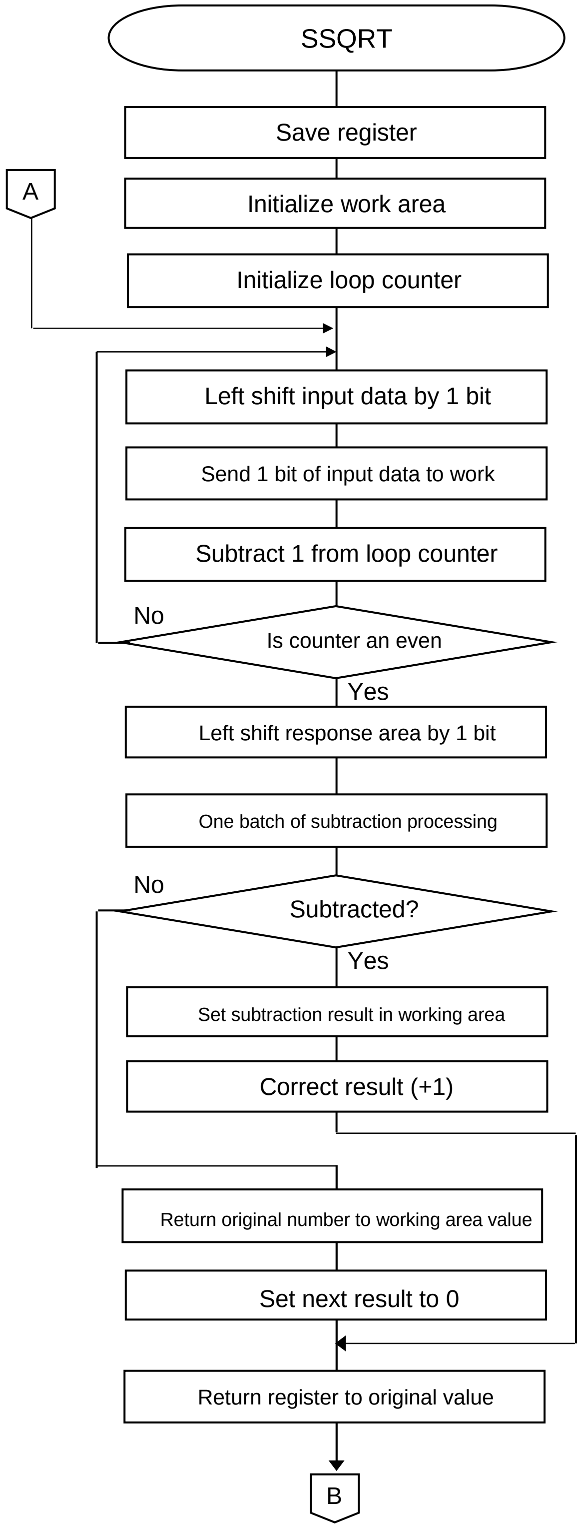
```
MOVW AX,    #0x3C44        ; Set lower-order word of the argument
MOVW BC,    #0x0017        ; Set higher-order word of the argument
CALL   !SSQRT              ; Square root
```

When using this program with C language, make sure you include the following prototype declaration:

```
uint16_t    _ssqrt(uint32_t)        ;
```

3.6 Program Flowcharts

Figure 3.5 and Figure 3.6 show the flowcharts for the square root operation.



Save HL and DE register data in stack.

Clear registers HL and DE and variables DIGITU1 and DIGITU2 to 0.

Set variable CLOOP0 to 32 as loop counter.

Shift AX register (lower bits) 1 bit; rotate through carry BC register (upper) 1 bit left.

Rotate through carry DE register 1 bit left.
Rotate through carry variable DIGITU1 1 bit left.

Decrement loop counter (shift 2 bits).

Set carry, then rotate through carry HL register 1 bit left.
Rotate through carry variable DIGITU2 1 bit left.

DIGITU1 and DE register - DIGITU2 and HL register
(Use AX register for subtraction.)

If working area is bigger than calculated-result area, set subtraction result to working area.

Add 1 to L register value.

AX register ← AX register + HL register

Subtract 1 from HL register value.

AX register ←→ DE register
This routine sets the working and calculation-result areas.

Figure 3.5 Square Root Operation Flowchart (1/2)

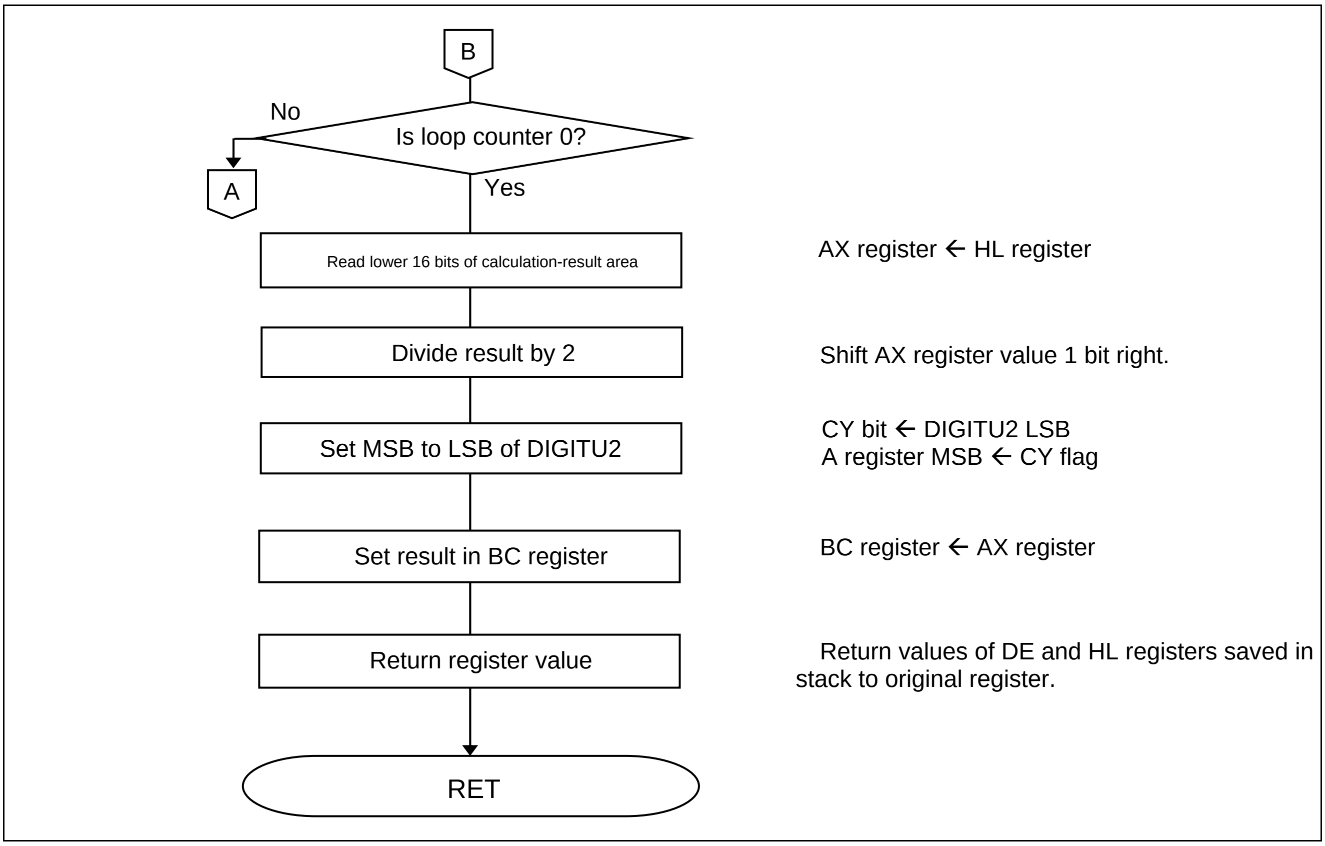


Figure 3.6 Square Root Operation Flowchart (2/2)

4. Sample Code

The sample code is available on the Renesas Electronics Website.

5. Documents for Reference

RL78/G10 User's Manual: Hardware (R01UH0384E)

RL78 Family User's Manual: Software (R01US0015E)

(The latest versions of the documents are available on the Renesas Electronics Website.)

Technical Updates/Technical Brochures

(The latest versions of the documents are available on the Renesas Electronics Website.)

Website and Support

Renesas Electronics Website

http://japan.renesas.com/

Inquiries

http://japan.renesas.com/contact/

Revision Record	RL78/G10 Square Root Program CC-RL

Rev.	Date	Description	
		Page	Summary
1.00	Feb. 03, 2016	—	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   ¾ The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   ¾ The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   ¾ The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   ¾ When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   ¾ The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0