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 .................................................................................................................................... 3
2. Operating Conditions ......................................................................................................................... 3
3. Software Explanation ........................................................................................................................ 4
   3.1 Operation Outline .......................................................................................................................... 4
   3.2 Square Root Operation Concept ................................................................................................... 4
   3.3 Library Functions (subroutines) .................................................................................................... 6
   3.4 Library Function (subroutine) Specifications .................................................................................. 6
   3.5 How to Use the Library Function (subroutine) .............................................................................. 6
   3.6 Program Flowcharts ....................................................................................................................... 7
4. Sample Code ..................................................................................................................................... 9
5. Documents for Reference ................................................................................................................. 9
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>
<thead>
<tr>
<th>Function</th>
<th>Operation description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square root operation</td>
<td>Inputs a 32-bit binary number and produces its square root as a 16-bit binary number.</td>
</tr>
</tbody>
</table>

2. Operating Conditions

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

Table 2.1 Operating Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>RL78/G10 (R5F10Y16)</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>High-speed on-chip oscillator clock (HOCO): 20MHz</td>
</tr>
<tr>
<td></td>
<td>CPU/peripheral hardware clock: 20MHz</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>5.0 V (can run on a voltage range of 2.9 V to 5.5 V.)</td>
</tr>
<tr>
<td></td>
<td>SPOR detection voltage: Rising edge voltage: 2.90V</td>
</tr>
<tr>
<td></td>
<td>: Falling edge voltage: 2.84V</td>
</tr>
<tr>
<td>Integrated development environment (CS+)</td>
<td>CS+ for CC V3.01.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Assembler (CS+)</td>
<td>CC-RL V1.01.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Integrated development environment (e² studio)</td>
<td>e² studio V4.0.2.008 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Assembler (e² studio)</td>
<td>CC-RL V1.01.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Usage environment</td>
<td>RL78/G10 simulator</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Register/Memory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AX register</td>
<td>Lower-order word of input data and shift work area</td>
</tr>
<tr>
<td>BC register</td>
<td>Higher-order word of input data and shift work area</td>
</tr>
<tr>
<td>DE register</td>
<td>Work area</td>
</tr>
<tr>
<td>HL register</td>
<td>Result area</td>
</tr>
<tr>
<td>DIGITU1</td>
<td>Work area extension</td>
</tr>
<tr>
<td>DIGITU2</td>
<td>Calculation-result area extension</td>
</tr>
</tbody>
</table>

![Figure 3.1 Register and Memory Configuration](image)

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).

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

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

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>
<thead>
<tr>
<th>Function Name (subroutine)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>__ssqrt</td>
<td>Square root operation for 32-bit data</td>
</tr>
</tbody>
</table>

3.4 Library Function (subroutine) Specifications

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

Function Name: __ssqrt

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

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:

```assembly
.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.

```assembly
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:

```c
uint16_t _ssqrt(uint32_t) ;
```
3.6 Program Flowcharts

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

![Square Root Operation Flowchart (1/2)](image)

**Figure 3.5** Square Root Operation Flowchart (1/2)
B

Is loop counter 0?

No

A

Read lower 16 bits of calculation-result area

Divide result by 2

Set MSB to LSB of DIGITU2

Set result in BC register

Return register value

RET

Yes

AX register ← HL register

Shift AX register value 1 bit right.

CY bit ← DIGITU2 LSB
A register MSB ← CY flag

BC register ← AX register

Return values of DE and HL registers saved in stack to original register.

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/
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Feb. 03, 2016</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>
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.
   - 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.
   - 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.
   - 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.
   - 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.
   - 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.
Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability or responsibility for any errors, omissions or malfunctions occurring in the information included herein.

3. Renesas Electronics' products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or your third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.

4. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall not have any liability for malfunctions or damages arising out of the use of Renesas Electronics' products beyond such specified range.

5. Renesas Electronics reserves the right to modify, alter, improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, the control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.

6. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you as a result of such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.

7. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or your third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall not have any liability for malfunctions or damages arising out of the use of Renesas Electronics' products beyond such specified range.

9. Renesas Electronics reserves the right to modify, alter, improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, the control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.

10. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you as a result of such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.

11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics products" in this document means any product developed or manufactured by or for Renesas Electronics.