User's Manual

32

RH850G4KH

User's Manual: Software

Renesas microcontroller

RH850 Family

Renesas Electronics
www.renesas.com

Rev.1.00 Sep. 2025

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

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. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to power supply or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.
5. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
6. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
7. Power ON/OFF sequence
   In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

Table of Contents

Section 1 Overview

1.1 Purpose of This User's Manual

This user's manual is intended to describe the details of instructions available in the RH850G4KH.

There are some variations of RH850G4x core. Those can be identified by PID value.

If there is a difference between variations, PID value is shown in this document. Other contents are common for all variations.

The supported RH850G4KH cores are described below.

RH850G4x Core name	PID[31:24] Bit Value
RH850G4KH	08_H

Note: For details of the RH850G4KH architecture, refer to the hardware manual of the product used.

This user's manual only includes the RH850G4KH specific description. The common description is described in the *RH850G4MH User's Manual: Software*.

Section 2 Instruction

2.1 Basic Instructions

2.1.1 Overview of Basic Instructions

(1) Special Instructions

This CPU does not support the following instructions.

- LDM.MP: Load Multiple MPU entries from memory (Supported only when Architecture Identifier bit PID[31:24] = 07_H (RH850G4MH2))
- STM.MP: Store Multiple MPU entries to memory (Supported only when Architecture Identifier bit PID[31:24] = 07_H (RH850G4MH2))

This CPU supports the synchronization for FPI exception using SYNCE.

- SYNCE: Synchronize exceptions

(2) Extended Floating-point Instructions

This CPU does not support the Extended Floating-point instructions.

2.1.2 Basic Instruction Set

2.1.2.1 SYNCE

<Special instruction>

Synchronize exceptions

SYNCE

Exception synchronization instruction

[Instruction format] SYNCE

[Operation] Waits for the synchronization of exceptions.

[Format] Format I

[Opcode]

15	0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1	

[Flags]

CY	—
OV	—
S	—
Z	—
SAT	—

[Description] Waits for the synchronization of all preceding exceptions before starting execution. It does not perform any operation but is completed when its execution is started. Exception synchronization means that all exceptions that are generated by the preceding instructions are notified to the CPU and are kept waiting until their priority is judged. If a condition of acknowledging exceptions is satisfied before this instruction is executed, therefore, all imprecise exceptions (FPI exceptions) that are generated because of the preceding instructions are always acknowledged before execution of this instruction is completed. This instruction can be used to guarantee completion of exception handling by the preceding task before a task is changed or terminated in a multi-processing environment.

If the target exceptions do not be supported in the CPU, SYNCE performs no specific processing that involves synchronization processing. When the execution of the SYNCE instruction is completed, the PC proceeds to the next instruction. However, an interrupt can be acknowledged.

Appendix A Number of Instruction Execution Clocks

A.1 Numbers of Clock Cycles for Execution

Numbers of clock cycles for execution are given in this section. Since the G4KH has a pipelined architecture that differs from that of other CPUs, the various values given cannot be treated in a uniform manner. Moreover, the number of clock cycles required to execute an actual instruction may differ with the state of execution of the previous and next instructions.

A.2 Number of Instruction Execution Clocks

Legend of Execution Clocks

Symbol	Description
issue	When the other instruction is executed immediately after the execution of the current instruction.
repeat	When the same instruction is repeated immediately after the execution of the current instruction.
latency	When the following instruction uses the result of the current instruction.

(1/7)

Types of Instructions	Mnemonics	Operand	Instruction Length (Number of Bytes)	Number of Execution Clocks		
				issue	repeat	latency
Load instruction	LD.B	disp16[reg1],reg2	4	1	1	2*1
	LD.B	disp23[reg1],reg3	6	1	1	2*1
	LD.B	[reg1]+,reg3	4	2	2	2*1
	LD.B	[reg1]−,reg3	4	2	2	2*1
	LD.BU	disp16[reg1],reg2	4	1	1	2*1
	LD.BU	disp23[reg1],reg3	6	1	1	2*1
	LD.BU	[reg1]+,reg3	4	2	2	2*1
	LD.BU	[reg1]−,reg3	4	2	2	2*1
	LD.DW	disp23[reg1],reg3	6	2	2	3*1
	LD.H	disp16[reg1],reg2	4	1	1	2*1
	LD.H	disp23[reg1],reg3	6	1	1	2*1
	LD.H	[reg1]+,reg3	4	2	2	2*1
	LD.H	[reg1]−,reg3	4	2	2	2*1
	LD.HU	disp16[reg1],reg2	4	1	1	2*1
	LD.HU	disp23[reg1],reg3	6	1	1	2*1
	LD.HU	[reg1]+,reg3	4	2	2	2*1
	LD.HU	[reg1]−,reg3	4	2	2	2*1
	LD.W	disp16[reg1],reg2	4	1	1	2*1
	LD.W	disp23[reg1],reg3	6	1	1	2*1
	LD.W	[reg1]+,reg3	4	2	2	2*1
	LD.W	[reg1]−,reg3	4	2	2	2*1
	SLD.B	disp7[ep],reg2	2	1	1	2*1
	SLD.BU	disp4[ep],reg2	2	1	1	2*1
	SLD.H	disp8[ep],reg2	2	1	1	2*1
	SLD.HU	disp5[ep],reg2	2	1	1	2*1
	SLD.W	disp8[ep],reg2	2	1	1	2*1

(2/7)

Types of Instructions	Mnemonics	Operand	Instruction Length (Number of Bytes)	Number of Execution Clocks		
				issue	repeat	latency
Store instruction	ST.B	reg2,disp16[reg1]	4	3	3	3*1
	ST.B	reg3,disp23[reg1]	6	3	3	3*1
	ST.B	reg3,[reg1]+	4	4	4	4*1
	ST.B	reg3,[reg1]−	4	4	4	4*1
	ST.DW	reg3,disp23[reg1]	6	2	2	2*1
	ST.H	reg2,disp16[reg1]	4	3	3	3*1
	ST.H	reg3,disp23[reg1]	6	3	3	3*1
	ST.H	reg3,[reg1]+	4	4	4	4*1
	ST.H	reg3,[reg1]−	4	4	4	4*1
	ST.W	reg2,disp16[reg1]	4	1	1	1*1
	ST.W	reg3,disp23[reg1]	6	1	1	1*1
	ST.W	reg3,[reg1]+	4	2	2	2*1
	ST.W	reg3,[reg1]−	4	2	2	2*1
	SST.B	reg2,disp7[ep]	2	3	3	3*1
	SST.H	reg2,disp8[ep]	2	3	3	3*1
	SST.W	reg2,disp8[ep]	2	1	1	1*1
Bit manipulation instruction	CLR1	bit#3,disp16[reg1]	4	3	3	3
	CLR1	reg2,[reg1]	4	3	3	3
	NOT1	bit#3,disp16[reg1]	4	3	3	3
	NOT1	reg2,[reg1]	4	3	3	3
	SET1	bit#3,disp16[reg1]	4	3	3	3
	SET1	reg2,[reg1]	4	3	3	3
	TST1	bit#3,disp16[reg1]	4	3	3	3
	TST1	reg2,[reg1]	4	3	3	3
Special instruction	CAXI	[reg1],reg2,reg3	4	4	4	5
	DISPOSE*4	imm5,list12	4	N+2*3	N+2*3	N+2*3
	LDL.BU	[reg1],reg3	6	1	1	2*1
	LDL.HU	[reg1],reg3	6	1	1	2*1
	LDL.W	[reg1],reg3	6	1	1	2*1
	POPSP	rh-rt	4	N+2*5	N+2*5	N+2*5
	LDM.MP	[reg1], eh-et	*18	*18	*18	*18
	PREPARE	list12,imm5	4	N+2*3	N+2*3	N+2*3
	PREPARE	list12,imm5,sp	4	N+3*3	N+3*3	N+3*3
	PREPARE	list12,imm5,imm16	6	N+3*3	N+3*3	N+3*3
	PREPARE	list12,imm5,imm16<<16	6	N+3*3	N+3*3	N+3*3
	PREPARE	list12,imm5,imm32	8	N+3*3	N+5*3	N+3*3
	PUSHSP	rh-rt	4	N+2*5	N+2*5	N+2*5
	RESBANK	—	4	N+3*16	N+3*16	N+3*16
	STC.B	reg3,[reg1]	6	1	1	2*1
	STC.H	reg3,[reg1]	6	1	1	2*1

(3/7)

Types of Instructions	Mnemonics	Operand	Instruction Length (Number of Bytes)	Number of Execution Clocks		
				issue	repeat	latency
Special instruction	STC.W	reg3,[reg1]	6	1	1	2*1
	STM.MP	eh-et, [reg1]	*18	*18	*18	*18
Multiplication instruction*21	MUL	reg1,reg2,reg3	4	2	2	4
	MUL	imm9,reg2,reg3	4	2	2	4
	MULH	reg1,reg2	2	1	1	3
	MULH	imm5,reg2	2	1	1	3
	MULHI	imm16,reg1,reg2	4	1	1	3
	MULU	reg1,reg2,reg3	4	2	2	4
	MULU	imm9,reg2,reg3	4	2	2	4
Multiply-accumulate operation instruction	MAC	reg1,reg2,reg3,reg4	4	3	3	5
	MACU	reg1,reg2,reg3,reg4	4	3	3	5
Arithmetic instruction	ADD	reg1,reg2	2	1	1	1
	ADD	imm5,reg2	2	1	1	1
	ADDI	imm16,reg1,reg2	4	1	1	1
	CMP	reg1,reg2	2	1	1	1
	CMP	imm5,reg2	2	1	1	1
	MOV	reg1,reg2	2	1	1	1
	MOV	imm5,reg2	2	1	1	1
	MOV	imm32,reg1	6	1	1	1
	MOVEA	imm16,reg1,reg2	4	1	1	1
	MOVHI	imm16,reg1,reg2	4	1	1	1
	SUB	reg1,reg2	2	1	1	1
	SUBR	reg1,reg2	2	1	1	1
Conditional operation instruction	ADF	cccc,reg1,reg2,reg3	4	1	1	1
	SBF	cccc,reg1,reg2,reg3	4	1	1	1
Saturated operation instruction	SATADD	reg1,reg2	2	1	1	1
	SATADD	imm5,reg2	2	1	1	1
	SATADD	reg1,reg2,reg3	4	1	1	1
	SATSUB	reg1,reg2	2	1	1	1
	SATSUB	reg1,reg2,reg3	4	1	1	1
	SATSUBI	imm16,reg1,reg2	4	1	1	1
	SATSUBR	reg1,reg2	2	1	1	1
Logical instruction	AND	reg1,reg2	2	1	1	1
	ANDI	imm16,reg1,reg2	4	1	1	1
	NOT	reg1,reg2	2	1	1	1
	OR	reg1,reg2	2	1	1	1
	ORI	imm16,reg1,reg2	4	1	1	1
	TST	reg1,reg2	2	1	1	1
	XOR	reg1,reg2	2	1	1	1
	XORI	imm16,reg1,reg2	4	1	1	1

(4/7)

Types of Instructions	Mnemonics	Operand	Instruction Length (Number of Bytes)	Number of Execution Clocks		
				issue	repeat	latency
Data operation instruction	BINS	reg1,pos,width,reg2	4	1	1	1
	BSH	reg2,reg3	4	1	1	1
	BSW	reg2,reg3	4	1	1	1
	CLIP.B	reg1,reg2	2	1	1	1
	CLIP.BU	reg1,reg2	2	1	1	1
	CLIP.H	reg1,reg2	2	1	1	1
	CLIP.HU	reg1,reg2	2	1	1	1
	CMOV	cccc,reg1,reg2,reg3	4	1	1	1
	CMOV	cccc,imm5,reg2,reg3	4	1	1	1
	HSH	reg2,reg3	4	1	1	1
	HSW	reg2,reg3	4	1	1	1
	ROTL	imm5,reg2,reg3	4	1	1	1
	ROTL	reg1,reg2,reg3	4	1	1	1
	SAR	reg1,reg2	4	1	1	1
	SAR	imm5,reg2	2	1	1	1
	SAR	reg1,reg2,reg3	4	1	1	1
	SASF	cccc,reg2	4	1	1	1
	SETF	cccc,reg2	4	1	1	1
	SHL	reg1,reg2	4	1	1	1
	SHL	imm5,reg2	2	1	1	1
	SHL	reg1,reg2,reg3	4	1	1	1
	SHR	reg1,reg2	4	1	1	1
	SHR	imm5,reg2	2	1	1	1
	SHR	reg1,reg2,reg3	4	1	1	1
	SXB	reg1	2	1	1	1
	SXH	reg1	2	1	1	1
	ZXB	reg1	2	1	1	1
	ZXH	reg1	2	1	1	1
Bit search instruction	SCH0L	reg2,reg3	4	1	1	1
	SCH0R	reg2,reg3	4	1	1	1
	SCH1L	reg2,reg3	4	1	1	1
	SCH1R	reg2,reg3	4	1	1	1
Division instruction*21	DIV	reg1,reg2,reg3	4	20	20	20
	DIVH	reg1,reg2	2	20	20	20
	DIVH	reg1,reg2,reg3	4	20	20	20
	DIVHU	reg1,reg2,reg3	4	20	20	20
	DIVU	reg1,reg2,reg3	4	20	20	20
High-speed divide operation instruction	DIVQ	reg1,reg2,reg3	4	N+4*6	N+4*6	N+4*6
	DIVQU	reg1,reg2,reg3	4	N+4*6	N+4*6	N+4*6

(5/7)

Types of Instructions	Mnemonics	Operand	Instruction Length (Number of Bytes)	Number of Execution Clocks		
				issue	repeat	latency
Branch instruction*7*8	Bcond	disp9	2	3	3	3
	Bcond	disp9 (Always)	4	3	3	3
	Bcond	disp17	4	3	3	3
	JARL	disp22,reg2	6	3	4	3
	JARL	disp32,reg1	4	3	3	3
	JARL	[reg1],reg3	2	3	3	3
	JMP	[reg1]	6	3	4	3
	JMP	disp32[reg1]	4	2	2	2
	JR	disp22	6	2	3	2
	JR	disp32	2	3	3	3
Loop instruction*8	LOOP	reg1,disp16	4	4	4	4
Special instruction (with branching)	CALLT	imm6	2	14	14	14
	CTRET	—	4	3	3	3
	DISPOSE	imm5,list12,[reg1]	4	N+2*9	N+2*9	N+2*9
	EIRET	—	4	3	3	3
	FERET	—	4	3	3	3
	FETRAP	vector	2	4	4	4
	RIE	—	4	4	4	4
	TRAP	vector5	4	4	4	4
	SWITCH	reg1	2	6	6	6
	SYSCALL	vector8	4	6	6	6
Special instruction	DI	—	4	1	1	1
	EI	—	4	1	1	1
	HALT	—	4	Undefined*17	Undefined*17	Undefined*17
	LDSR	reg2,regID,selID	4	1*10	1*10	1
	NOP	—	2	1	1	1
	SNOOZE	—	4	Undefined*11	Undefined*11	Undefined*11
	STSR	regID,reg2,selID	4	1*10	1*10	2
	SYNCE	—	2	Undefined	Undefined	Undefined
	SYNCI	—	2	Undefined*12	Undefined*12	Undefined*12
	SYNCM	—	2	Undefined*13	Undefined*13	Undefined*13
	SYNCP	—	2	Undefined*14	Undefined*14	3
Cache instruction	CACHE	cacheop,[reg1]	4	*15	*15	*15
	PREF	prefop,[reg1]	4	*15	*15	*15

(6/7)

Types of Instructions	Mnemonics	Operand	Instruction Length (Number of Bytes)	Number of Execution Clocks (imprecise)			Number of Execution Clocks (precise)		
				issue	repeat	latency	issue	repeat	latency
Floating-point arithmetic operation (single precision) *21	ABSF.S	reg2,reg3	4	1	1	5	6	6	6
	ADDF.S	reg1,reg2,reg3	4	1	1	5	6	6	6
	CEILF.SL	reg2,reg3	4	2*19	2	6	6	6	6
	CEILF.SUL	reg2,reg3	4	2*19	2	6	6	6	6
	CEILF.SUW	reg2,reg3	4	1	1	5	6	6	6
	CEILF.SW	reg2,reg3	4	1	1	5	6	6	6
	CMOVF.S	cc,reg1,reg2,reg3	4	1	1	5	6	6	6
	CMPF.S	cond,reg1,reg2,cc	4	1	1	4	6	6	6
	CVTF.HS	reg2,reg3	4	1	1	5	6	6	6
	CVTF.LS	reg2,reg3	4	1	1	5	6	6	6
	CVTF.SH	reg2,reg3	4	1	1	5	6	6	6
	CVTF.SL	reg2,reg3	4	2	2	6	6	6	6
	CVTF.SUL	reg2,reg3	4	2	2	6	6	6	6
	CVTF.SUW	reg2,reg3	4	1	1	5	6	6	6
	CVTF.SW	reg2,reg3	4	1	1	5	6	6	6
	CVTF.ULS	reg2,reg3	4	1	1	5	6	6	6
	CVTF.UWS	reg2,reg3	4	1	1	5	6	6	6
	CVTF.WS	reg2,reg3	4	1	1	5	6	6	6
	DIVF.S	reg1,reg2,reg3	4	14	14	18	19	19	19
	FLOORF.SL	reg2,reg3	4	2	2	6	6	6	6
	FLOORF.SUL	reg2,reg3	4	2	2	6	6	6	6
	FLOORF.SUW	reg2,reg3	4	1	1	5	6	6	6
	FLOORF.SW	reg2,reg3	4	1	1	5	6	6	6
	FMAF.S	reg1,reg2,reg3	4	2	2	6	7	7	7
	FMSF.S	reg1,reg2,reg3	4	2	2	6	7	7	7
	FNMAF.S	reg1,reg2,reg3	4	2	2	6	7	7	7
	FNMSF.S	reg1,reg2,reg3	4	2	2	6	7	7	7
	MAXF.S	reg1,reg2,reg3	4	1	1	5	6	6	6
	MINF.S	reg1,reg2,reg3	4	1	1	5	6	6	6
	MULF.S	reg1,reg2,reg3	4	1	1	5	6	6	6
	NEGF.S	reg2,reg3	4	1	1	5	6	6	6
	RECIPF.S	reg2,reg3	4	10	10	14	15	15	15
	ROUNDF.SL	reg2,reg3	4	2	2	6	6	6	6
	ROUNDF.SUL	reg2,reg3	4	2	2	6	6	6	6
	ROUNDF.SUW	reg2,reg3	4	1	1	5	6	6	6
	ROUNDF.SW	reg2,reg3	4	1	1	5	6	6	6
	RSQRTF.S	reg2,reg3	4	14	14	18	19	19	19
	SQRTF.S	reg2,reg3	4	14	14	18	19	19	19
	SUBF.S	reg1,reg2,reg3	4	1	1	5	6	6	6
	TRFSR	cc	4	1	1	1	1	1	1
	TRNCF.SL	reg2,reg3	4	2	2	6	6	6	6
	TRNCF.SUL	reg2,reg3	4	2	2	6	6	6	6
	TRNCF.SUW	reg2,reg3	4	1	1	5	6	6	6
	TRNCF.SW	reg2,reg3	4	1	1	5	6	6	6

(7/7)

Types of Instructions	Mnemonics	Operand	Instruction Length (Number of Bytes)	Number of Execution Clocks (imprecise)			Number of Execution Clocks (precise)		
				issue	repeat	latency	issue	repeat	latency
Floating-point arithmetic operation (double precision)*21	ABSF.D	reg2,reg3	4	2*19	2	6	6	6	6
	ADDF.D	reg1,reg2,reg3	4	3*20	3	7	7	7	7
	CEILF.DL	reg2,reg3	4	2*19	2	6	6	6	6
	CEILF.DUL	reg2,reg3	4	2*19	2	6	6	6	6
	CEILF.DUW	reg2,reg3	4	1	1	5	6	6	6
	CEILF.DW	reg2,reg3	4	1	1	5	6	6	6
	CMOVF.D	cc,reg1,reg2,reg3	4	3*20	3	7	7	7	7
	CMPF.D	cond,reg1,reg2,cc	4	2	2	5	7	7	7
	CVTF.DL	reg2,reg3	4	2*19	2	6	6	6	6
	CVTF.DS	reg2,reg3	4	1	1	5	6	6	6
	CVTF.DUL	reg2,reg3	4	2*19	2	6	6	6	6
	CVTF.DUW	reg2,reg3	4	1	1	5	6	6	6
	CVTF.DW	reg2,reg3	4	1	1	5	6	6	6
	CVTF.LD	reg2,reg3	4	2*19	2	6	6	6	6
	CVTF.SD	reg2,reg3	4	2*19	2	6	6	6	6
	CVTF.ULD	reg2,reg3	4	2*19	2	6	6	6	6
	CVTF.UWD	reg2,reg3	4	2*19	2	6	6	6	6
	CVTF.WD	reg2,reg3	4	2*19	2	6	6	6	6
	DIVF.D	reg1,reg2,reg3	4	32*20	32	36	36	36	36
	FLOORF.DL	reg2,reg3	4	2*19	2	6	6	6	6
	FLOORF.DUL	reg2,reg3	4	2*19	2	6	6	6	6
	FLOORF.DUW	reg2,reg3	4	1	1	5	6	6	6
	FLOORF.DW	reg2,reg3	4	1	1	5	6	6	6
	MAXF.D	reg1,reg2,reg3	4	3*20	3	7	7	7	7
	MINF.D	reg1,reg2,reg3	4	3*20	3	7	7	7	7
	MULF.D	reg1,reg2,reg3	4	4*20	4	8	8	8	8
	NEGF.D	reg2,reg3	4	2*19	2	6	6	6	6
	RECIPF.D	reg2,reg3	4	27*19	27	31	31	31	31
	ROUNDF.DL	reg2,reg3	4	2*19	2	6	6	6	6
	ROUNDF.DUL	reg2,reg3	4	2*19	2	6	6	6	6
	ROUNDF.DUW	reg2,reg3	4	1	1	5	6	6	6
	ROUNDF.DW	reg2,reg3	4	1	1	5	6	6	6
	RSQRTF.D	reg2,reg3	4	37*19	37	41	41	41	41
	SQRTF.D	reg2,reg3	4	31*19	31	35	35	35	35
	SUBF.D	reg1,reg2,reg3	4	3*20	3	7	7	7	7
	TRNCF.DL	reg2,reg3	4	2*19	2	6	6	6	6
	TRNCF.DUL	reg2,reg3	4	2*19	2	6	6	6	6
	TRNCF.DUW	reg2,reg3	4	1	1	5	6	6	6
	TRNCF.DW	reg2,reg3	4	1	1	5	6	6	6

Note 1. If there are no wait states (cycles of waiting) associated with the memory access.

Note 2. If there are no wait states (cycles of waiting) associated with the memory access.

Note 3. N depends on the total number of registers specified as list12.

Note 4. This is the value of DISPOSE without branching. For details of DISPOSE with branching, refer to [Special instruction (with branching)].

Note 5. N depends on the total number of registers specified as rh-rt.

Note 6. N = int (((Number of bits in the absolute value of the dividend) - (Number of bits in the absolute value of the divisor)) / 2) + 1.
However, when N < 1, N becomes 1, except in the case of division by zero, where N becomes 0. The range of N is 0 to 16.

Note 7. The number of clock cycles for branch instruction.

Note 8. The number of clock cycles for Taken branch instruction.

Note 9. Refer to Note 3 for the value of N.

Note 10. SelID = 0, 1, 2, 3, 5 (however, regID is 15 or less). Access to system register 10, 13 stops the issuing of subsequent instruction. Otherwise, operation is with “issue” = 1, “repeat” = 1.

Note 11. Depends on the setting for operation of the SNOOZE instruction.

Note 12. Performs processing to synchronize instruction fetching.

Note 13. Performs processing to synchronize memory access.

Note 14. Performs processing to synchronize pipeline.

Note 15. The instruction execution is completed, but completion of the internal processing depends on the internal state of the instruction fetching unit.

Note 16. The number of registers in the execution list.

Note 17. After waiting like the one due to the SYNCM instruction has finished, the execution changes to a HALT state.

Note 18. G4KH does not support LDM/MP, STM.MP.

Note 19. For non-FPU instructions, issue is 1.

Note 20. For non-FPU instructions, issue is 2.

Note 21. The latency may become minus 1 depend on the operand register number if the instruction has multiple destination registers.

RH850G4KH User’s Manual: Software

Publication Date: Rev.0.10 August 31, 2022
Rev.1.00 September 30, 2025

Published by: Renesas Electronics Corporation

RH850G4KH

Renesas Electronics Corporation

R01US0598EJ0100