To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of 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. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may 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.
6. 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 whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "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.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
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 have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to 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, 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, please evaluate the safety of the final products or system 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
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 product(s)" means any product developed or manufactured by or for Renesas Electronics.

H8S/20103, H8S/20203, and H8S/20223 Groups

Using the ELC to Perform Operation with Three Events Connected

Introduction

The event link controller (ELC) embedded in the H8S/20103, H8S/20203, and H8S/20223 Groups is set up so that a receive-data-full signal from SCI3_1 starts timer RD_0 channel 0 and the compare match signal for a match between timer RD_0 channel 0 and GRA_0 starts A/D conversion, both without CPU intervention.

Target Devices

H8S/20103 (R4F20103)

H8S/20203 (R4F20203)

H8S/20223 (R4F20223)

Frequency Used in Confirming Operation

System clock $\phi = \phi osc = 20$ MHz

Contents

1. Specifications

Specifications of this sample task are as given below. Figure 1 shows an overview of how the ELC is used to set up operation with three events connected and figure 2 shows a schematic view of how this is employed.

1. Voltage to be measured is applied to the PB0/AN0 pin.
2. The PB0 pin is set to operate as the AN0 pin for analog input and the A/D conversion end interrupt is enabled.
3. Settings are made so that A/D conversion is initiated by the compare-match A signal from channel 0 of timer RD_0 as the event signal (so the sampling period for A/D conversion becomes the period of the compare-match signal).
4. Settings are made to select ϕ/32 as the clock source for channel 0 of Timer RD_0 and the compare-match signal for matches between timer RD_0 and GRA_0 as the signal that clears TRDCNT_0.
5. The setting is made so that channel 0 of timer RD_0 starts counting in response to event input.
6. Settings are made to select the receive-data-full signal from SCI3_1 as the event signal that starts counting by channel 0 of timer RD_0.
7. Reception and receive interrupts are enabled by setting SCI3_1 for communications in asynchronous mode at a bit rate of 9,600 bps with even parity and 1 stop bit.
8. Event input is enabled.
9. The I bit is cleared to enable interrupts.
10. After the receive-data-full signal of SCI3_1 has been generated, counting starts on channel 0 of timer RD_0 (until reception by SCI3_1 is disabled after eight bytes have been received).
11. Every time the compare match signal for a match between timer RD_0 channel 0 and GRA_0 is generated, A/D conversion of the level on the AN0 pin starts without CPU intervention.
12. Results of A/D conversion are sampled ten times, and the average of the eight results that exclude the minimum and maximum values is taken as valid.

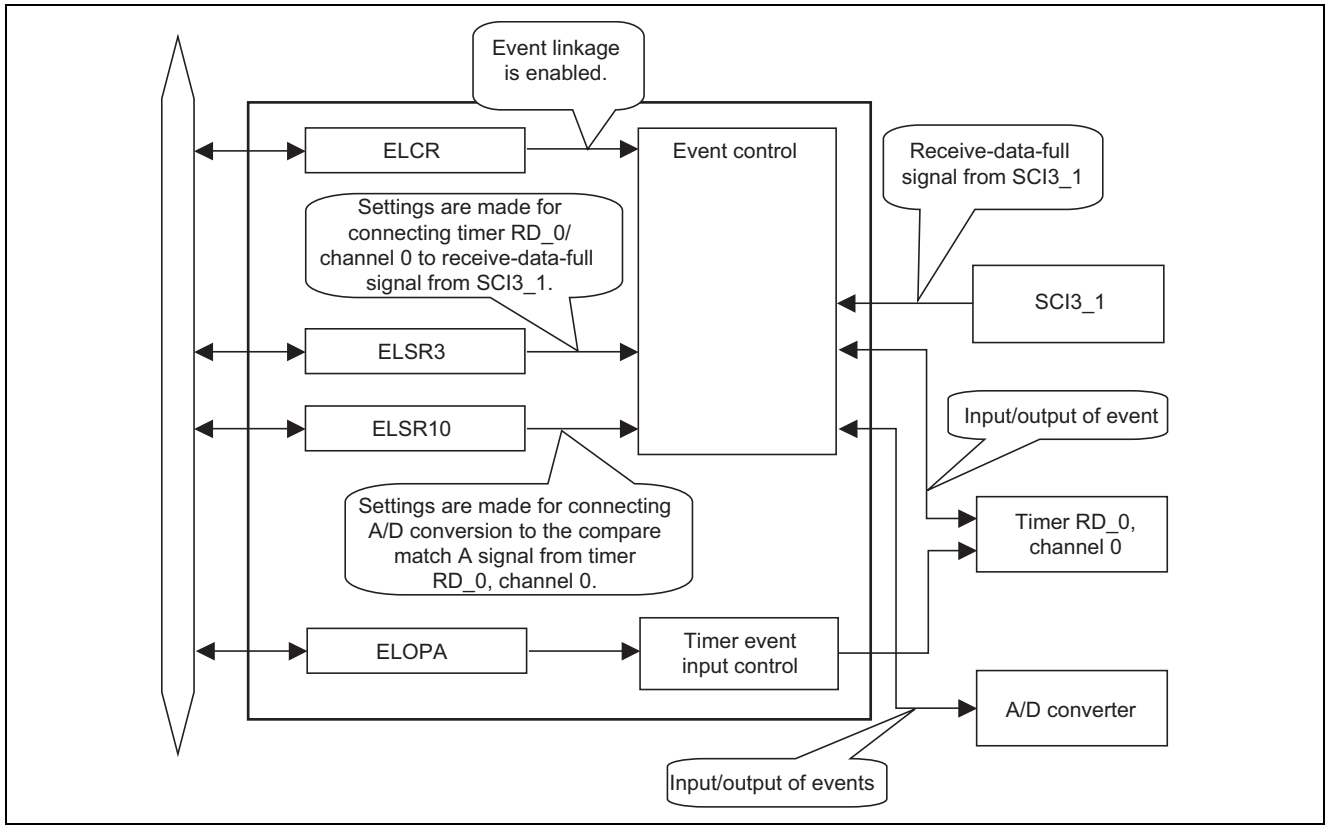


Figure 1 Overview of Using the ELC to Set up Operation with Three Events Connected

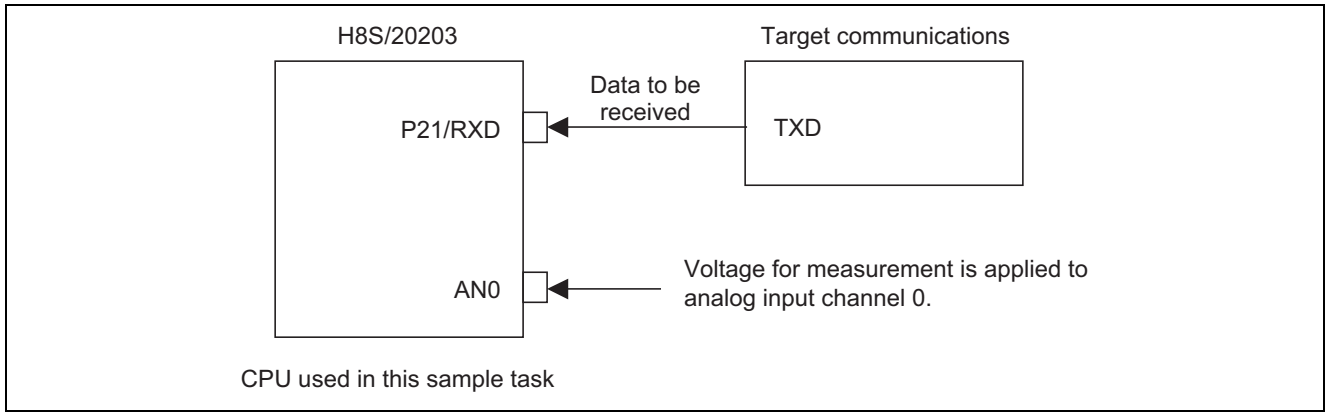


Figure 2 Schematic View of How Operation with Three Events Connected is Employed

2. Description of Modules Used

2.1 Event Link Controller (ELC)

The features of the ELC are described below. Figure 3 is a block diagram of the ELC.

The ELC connects events generated by the various peripheral modules to other modules. This function allows direct cooperation between modules, without CPU intervention.

- Fifty-nine event signals can be directly connected to modules.
- The operation of timer modules can be selected when an event is input to the timer module.
- Events can be connected to ports 3 and 6.
- Settings for ports enable the generation of events in the form of signals on port pins.
- A single bit or any grouping of several bits can be set up for event connection on the ports used for connecting events.
- The event generation timer can be used to set up the generation of signals on four channels as events with the desired intervals.

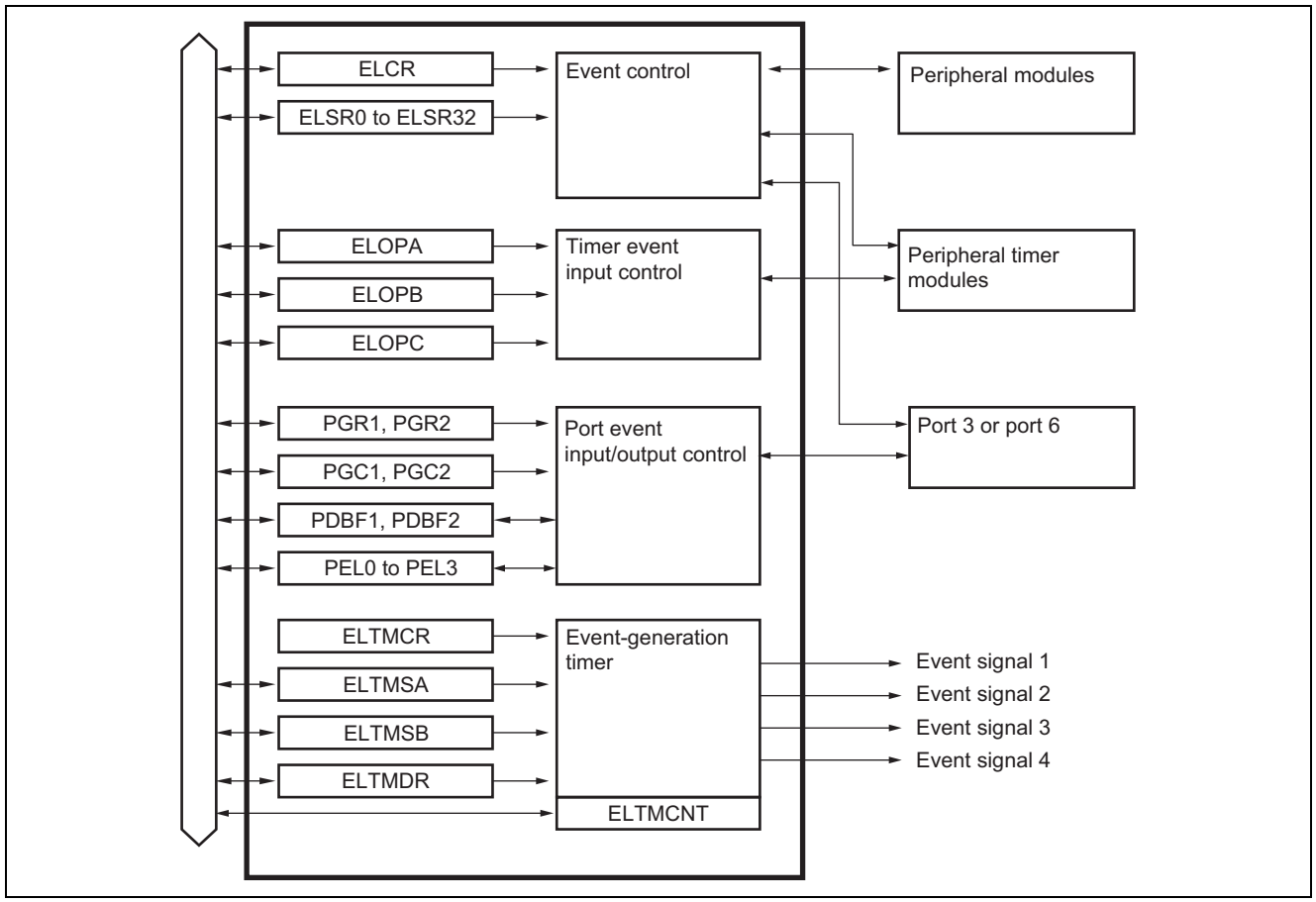


Figure 3 Block Diagram of Event Link Controller

2.1.1 Operation of Peripheral Timer Modules at the Time of Event Input

Timer modules may perform any of three operations in response to the input of a signal indicating an event (event signal below). The operation depends on the ELOP settings.

(1) Starting the Timer Counter
When the event signal is input, the count start bit* in the given timer control register is set to 1 to make the timer start counting. Input of the event signal while the count start bit is 1 is ineffective.
(2) Counting Events
The event signal is selected as the clock source for the timer so that the timer counts the events.
(3) Input Capture
Input of the event signal makes the timer perform input-capture operation.

Note: * See the descriptions of the bits in the relevant sections on timers.

2.1.2 Operation of A/D and D/A Converters at the Time of Event Input

Event signals are capable of setting the start bits* in A/D control registers and the output enable bits* in D/A control registers to 1, thus starting A/D or D/A conversion.

Note: * See the descriptions of the bits in the relevant sections on A/D and D/A converters.

2.2 Serial Communications Interface 3 (SCI3)

This LSI includes a serial communications interface 3 (SCI3), which has three independent channels. The SCI3 can handle both asynchronous and clock synchronous serial communications. In asynchronous mode, serial data communications can be carried out using standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communications Interface Adapter (ACIA). A function is also provided for serial communications between processors (multiprocessor communications function).

Table 1 shows the SCI3 channel configuration and figure 4 and figure 5 shows a block diagram of the SCI3. Since pin functions are identical for each of the three channels (SCI3, SCI3_2, and SCI3_3), separate explanations are not given in this section.

- Choice of asynchronous or clock synchronous serial communications modes
- Full-duplex communications capability
  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously. Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception.
- On-chip baud rate generator allows any bit rate to be selected
- External clock or on-chip baud rate generator can be selected as a transfer clock source.
- Six interrupt sources
  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, and parity error. The DTC can be activated by the transmit-data-empty interrupt and receive-data-full interrupt sources.
- High-speed access by the internal 16-bit bus
  16-bit TRDCNT and GR registers can be accessed in high speed by a 16-bit bus interface

Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: A break can be detected by reading the RXD pin level directly in the case of a framing error.

Clock synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors

Table 1 Configuration of SCI3 Channels

Channel	Abbreviation	Pin	Register	Register Address	Noise Canceller
Channel 1	SCI3*[1]	SCK3 RXD TXD	SMR	H'FF0550	Available
			BRR	H'FF0551	
			SCR3	H'FF0552	
			TDR	H'FF0553	
			SSR	H'FF0554	
			RDR	H'FF0555	
			RSR	—	
			TSR	—	
			SPMR	H'FF0556	
Channel 2	SCI3_2*[2]	SCK3_2 RXD_2/IrRxD TXD_2/IrTxD	SMR_2	H'FF0558	Available
			BRR_2	H'FF0559	
			SCR3_2	H'FF055A	
			TDR_2	H'FF055B	
			SSR_2	H'FF055C	
			RDR_2	H'FF055D	
			RSR_2	—	
			TSR_2	—	
			SPMR_2	H'FF055E	
			IrCR	H'FF05DE	
Channel 3	SCI3_3	SCK3_3 RXD_3 TXD_3	SMR_3	H'FF0560	Available
			BRR_3	H'FF0561	
			SCR3_3	H'FF0562	
			TDR_3	H'FF0563	
			SSR_3	H'FF0564	
			RDR_3	H'FF0565	
			RSR_3	—	
			TSR_3	—	
			SPMR_3	H'FF0566	

Notes: 1. Channel 1 of the SCI3 is used with boot mode as the on-board programming mode.
2. SCI3_2 is capable of transmitting and receiving IrDA (Infrared Data Association) communications waveforms based on IrDA standard version 1.0.

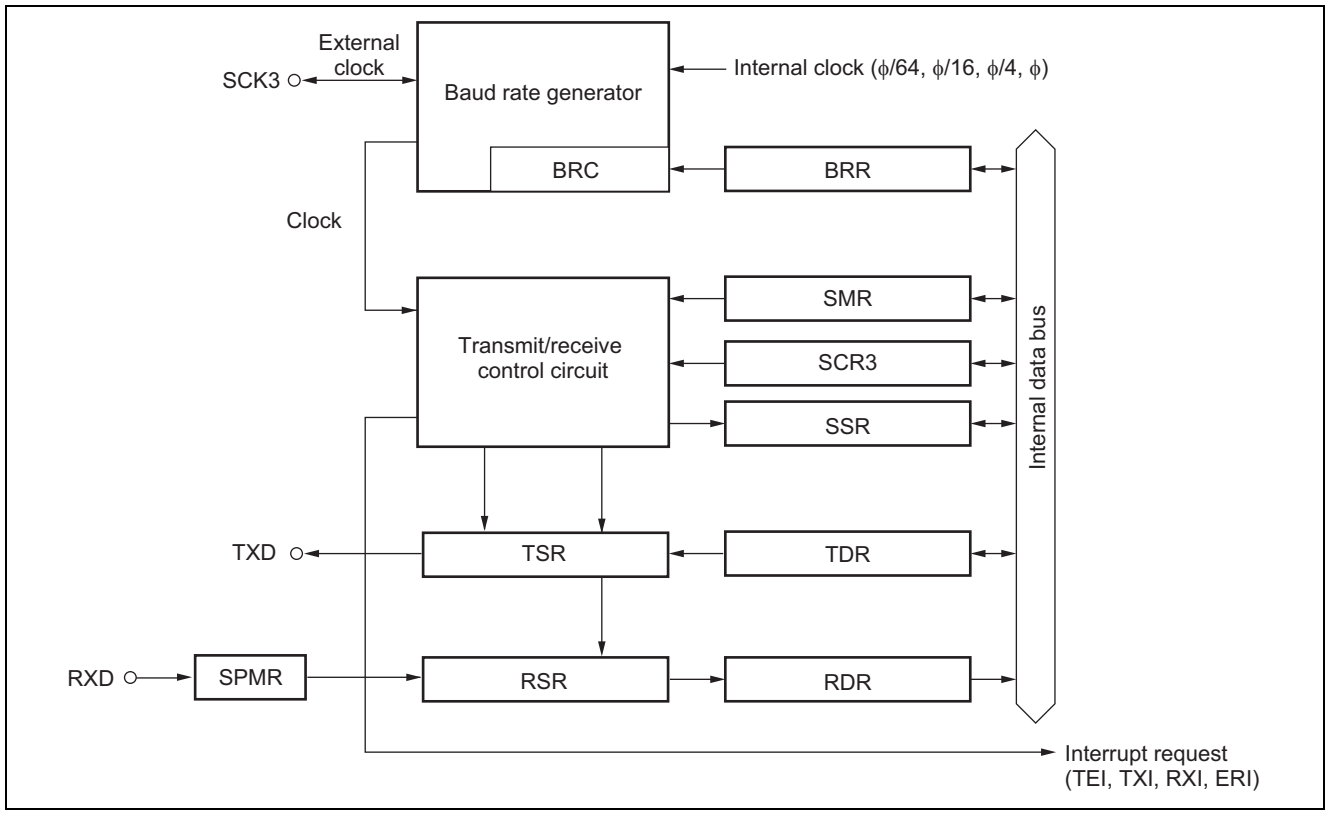


Figure 4 Block Diagram of SCI3 and SCI3_3

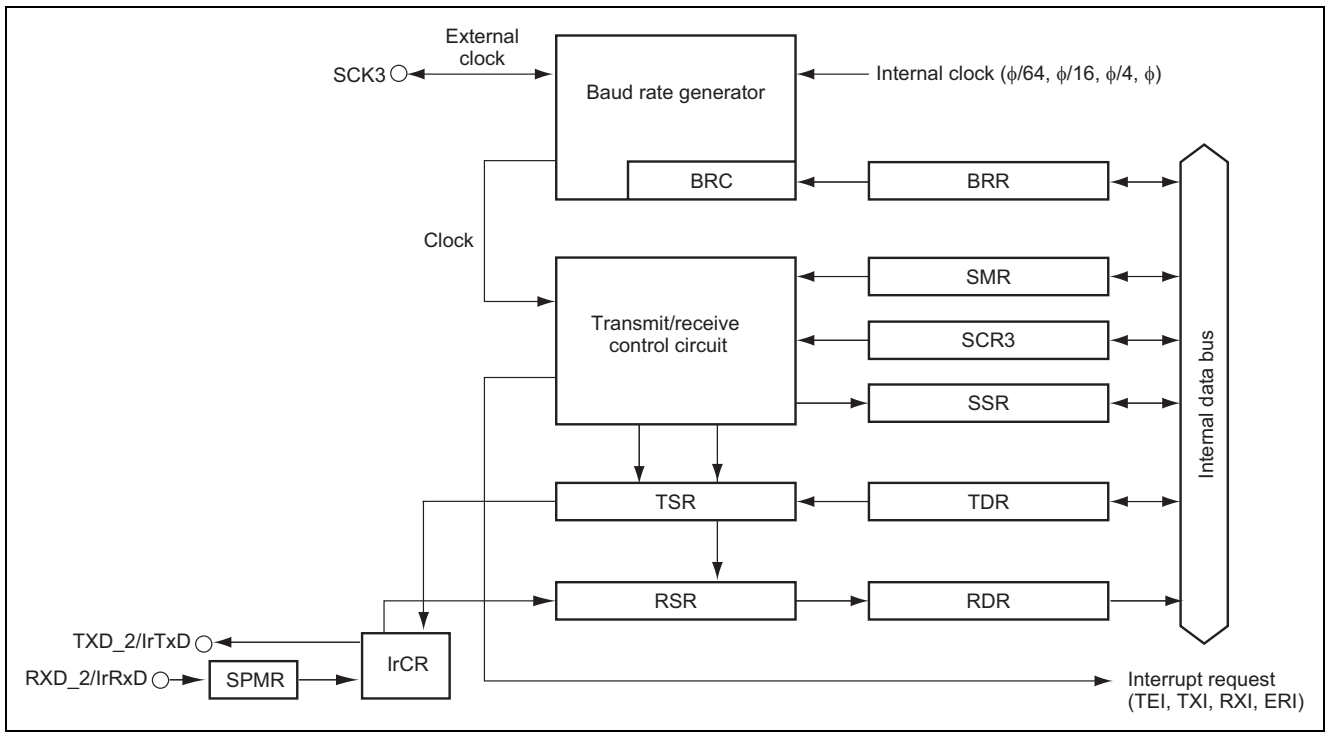


Figure 5 Block Diagram of SCI3_2

2.3 Timer RD

The features of timer RD are described below.

This LSI has two units of 16-bit timers (timer RD_0 and timer RD_1), each of which has two channels. Table 2 lists the timer RD functions, table 3 lists the channel configuration of timer RD, and figure 6 is a block diagram of the entire timer RD. Block diagrams of channels 0 and 1 are shown in figures 7 and 8.

Timer RD_0 has the same features as timer RD_1. Therefore, the unit number (_0 or _1) is not explicitly mentioned in this section unless otherwise noted (products of the H8S/20103 Group do not include timer RD_1).

- Capability to process up to eight inputs/outputs
- Eight general registers (GR): four registers for each channel
  Independently assignable output compare or input capture functions
- Selection of seven counter clock sources:
  Six internal clock signals (with the high-speed on-chip oscillator operating at 32 or 40 MHz: ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/32, and ϕ40) and an external clock
- Seven selectable operating modes
  - Timer mode
    Output compare function (Selection of 0 output, 1 output, or toggled output)
    Input capture function (Rising edge, falling edge, or both edges)
  - Synchronous operation
    Timer counters_0 and _1 (TRDCNT_0 and TRDCNT_1) can be written simultaneously.
    Simultaneous clearing by compare match or input capture is possible.
  - PWM mode
    Up to six-phase PWM output can be provided with desired duty ratio.
  - PWM3 mode
    One-phase PWM output for non-overlapped normal and counter phases
  - Reset synchronous PWM mode
    Three-phase PWM output for normal and counter phases
  - Complementary PWM mode
    Three-phase PWM output for non-overlapped normal and counter phases
    The A/D conversion start trigger can be set for PWM cycles.
  - Buffer operation
    The input capture registers can be configured for buffered operation.
    The output compare register can automatically be modified.
- High-speed access by the internal 16-bit bus
  16-bit TRDCNT and GR registers can be accessed in high speed by a 16-bit bus interface.
- Any initial timer output value can be set.
- Output of the timer can be disabled by an external trigger.
- Eleven interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt are available for each channel. An underflow interrupt can be set for channel 1.

Table 2 Functions of Timer RD (One Unit)

Item		Channel 0	Channel 1
Count clock		Internal clocks: φ, φ/2, φ/4, φ/8, φ/32, φ/40 External clock: FTIOA0 (TCLK)	
General registers (output compare/input capture registers)		GRA_0, GRB_0, GRC_0, GRD_0	GRA_1, GRB_1, GRC_1, GRD_1
Buffer registers		GRC_0, GRD_0	GRC_1, GRD_1
I/O pins		FTIOA0, FTIOB0, FTIOC0, FTIOD0	FTIOA1, FTIOB1, FTIOC1, FTIOD1
Counter clearing function		Compare match/input capture of GRA_0, GRB_0, GRC_0, or GRD_0	Compare match/input capture of GRA_1, GRB_1, GRC_1, or GRD_1
Compare match output	0 output	Yes	Yes
	1 output	Yes	Yes
	Toggled output	Yes	Yes
Input capture function		Yes	Yes
Synchronous operation		Yes	Yes
PWM mode		Yes	Yes
PWM3 mode		Yes	Yes
Reset-synchronized PWM mode		Yes	Yes
Complementary PWM mode		Yes	Yes
Buffer function		Yes	Yes
Interrupt sources		Compare match/ input capture A0 to D0 Overflow	Compare match/ input capture A1 to D1 Overflow Underflow

Table 3 Configuration of Timer RD Channels

Unit	Channel	Pin
Timer RD_0 (Unit 0)	0	FTIOA0
		FTIOB0
		FTIOC0
		FTIOD0
	1	FTIOA1
		FTIOB1
		FTIOC1
		FTIOD1
	Shared by channels 0 and 1	$\overline{\text{TRDOI\_0}}$
Timer RD_1 (Unit 1)	2	FTIOA2
		FTIOB2
		FTIOC2
		FTIOD2
	3	FTIOA3
		FTIOB3
		FTIOC3
		FTIOD3
	Shared by channels 2 and 3	$\overline{\text{TRDOI\_1}}$

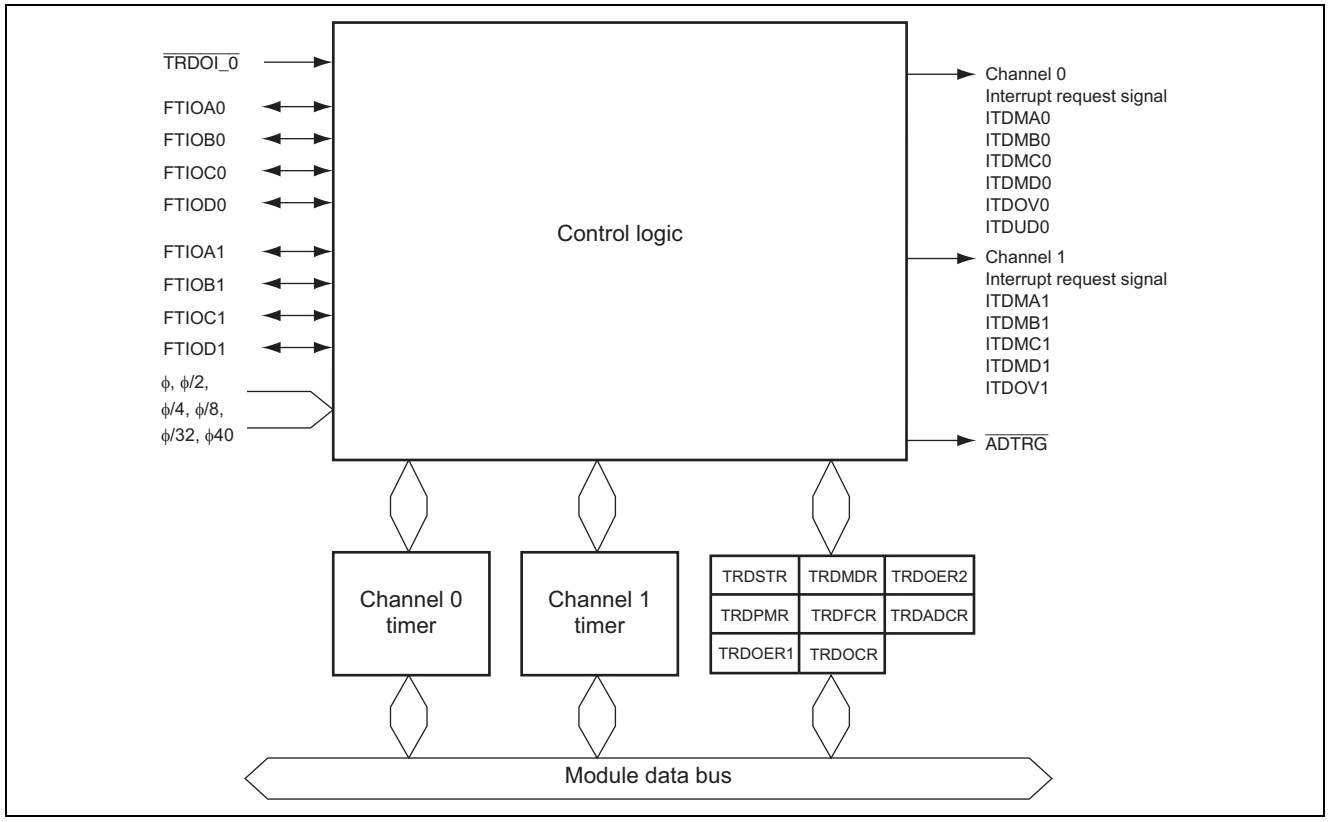


Figure 6 Block Diagram of Timer RD (One Unit)

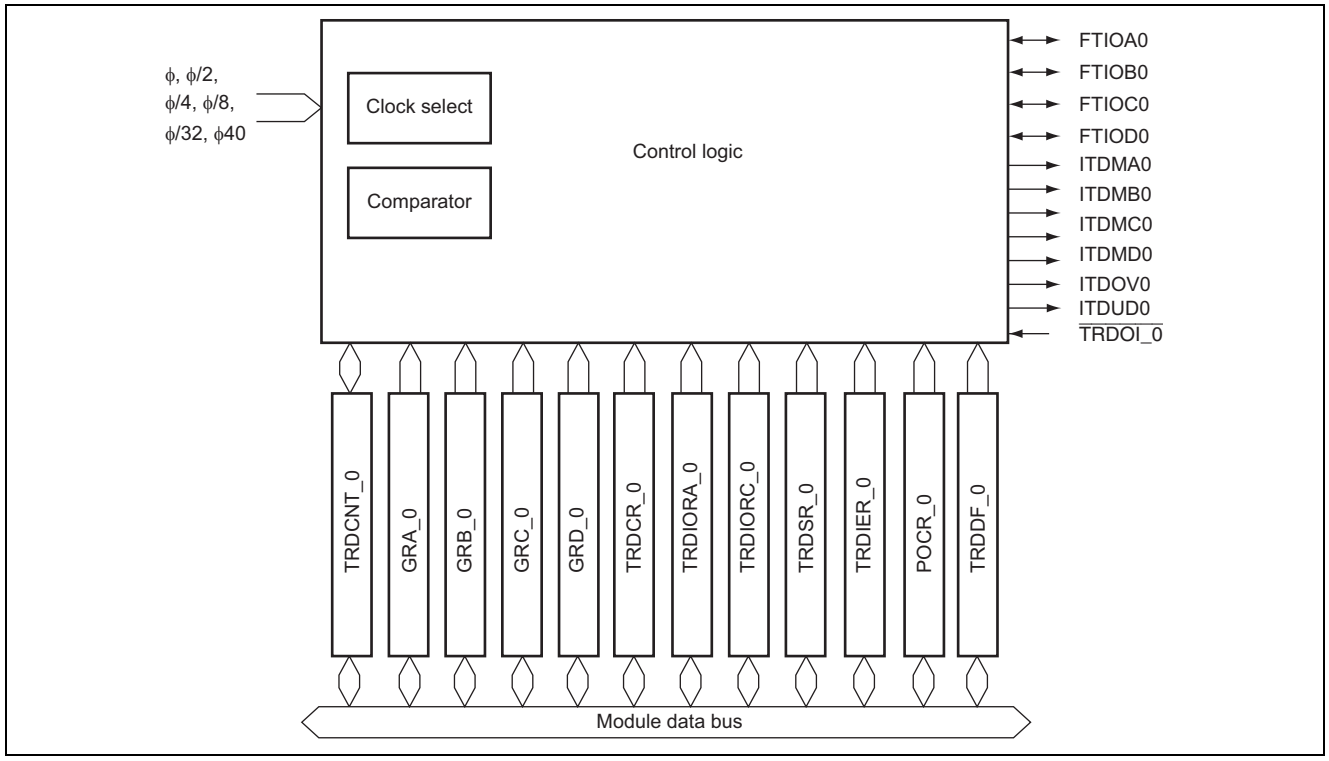


Figure 7 Block Diagram of Timer RD (Channel 0)

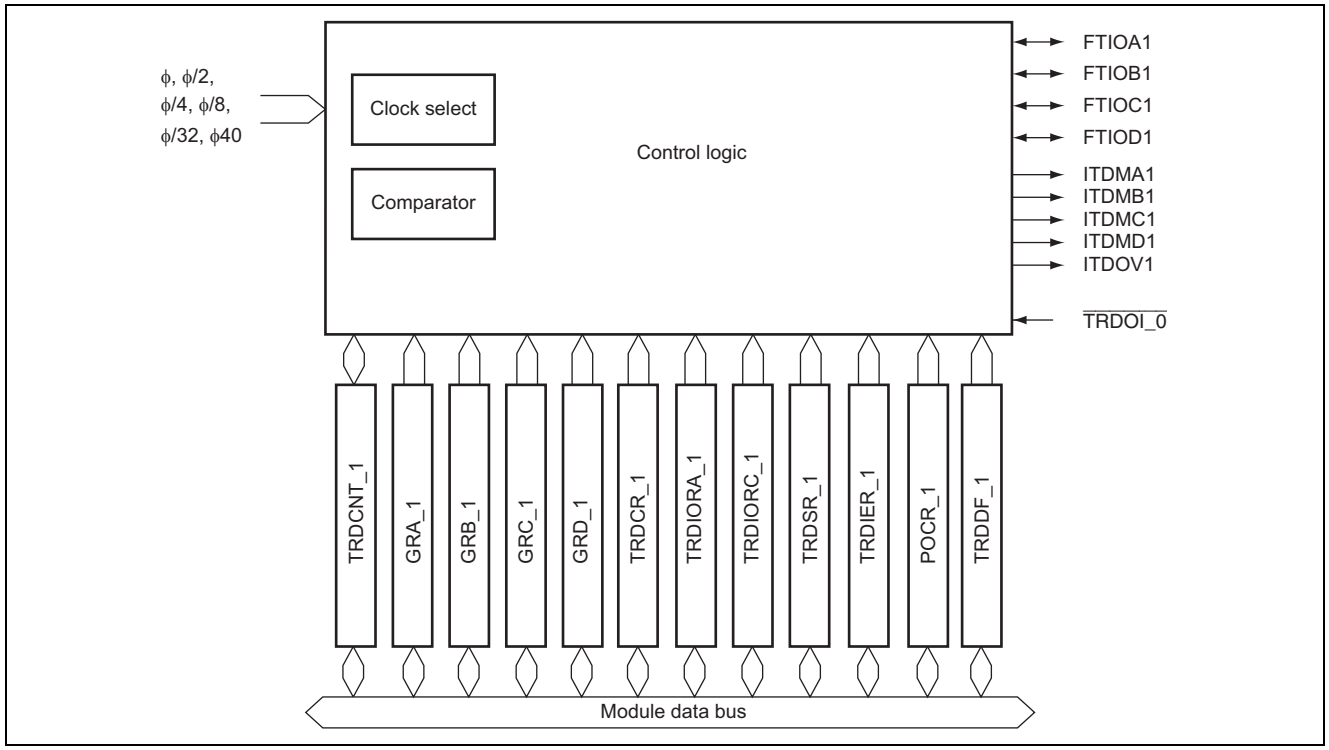


Figure 8 Block Diagram of Timer RD (Channel 1)

2.3.1 Operation Controlled by Event Links

Using the event link controller (ELC), timer RD unit 0 can be made to operate in the following ways in relation to events occurring in other modules. Each channel 0 and 1 can be specified independently.

1. Starting Counter Operation
   The start of counting operations by timer RD can be selected by ELOPA and ELOPB of the ELC. When the event specified by ELSR3 and ELSR4 occurs, the STR[1:0] bits in TRDSTR are set to 1, which stars counting by timer RD. However, if the specified event occurs when the STR bit has already been set to 1, the event is ineffective.

2. Counting Instances of an Event
   The counting of events by timer RD can be selected by ELOPA and ELOPB of the ELC. When the event specified in ELSR3 and ELSR4 occurs, event counter operation proceeds with that event as the source to drive counting, regardless of the setting of the TPSC[2:0] bits in TRDCR1. When the value of the counter is read, the value read out is the actual number of input events.

3. Input Capture
   Input capture operation of timer RD can be selected by ELOPA and ELOPB of the ELC. When the event specified in ELSR3 and ELSR4 occurs, GRD captures the value of TRDCNT. When input capture operation initiated by an event link is in use, set the IOD[3:0] bits = B'1101 in TRDIORC of timer RD, set the STR bit in TRDSTR to 1, and then start the counter. Since input on the FTIOD pin becomes valid at the same time, fix the input to the FTIOD pin or take other measures such as not allocating the FTIOD pin to the port in the PMC, etc.

2.4 A/D Converter

The features of the A/D converter are described below.

This LSI includes a successive approximation type 10-bit A/D converter (one unit or two units) that allows up to sixteen analog input channels to be selected. Figures 9 and 10 show the block diagrams of A/D converters unit 1 and unit 2, respectively.

The differences between unit 1 and unit 2 are the number of analog input channels and the number of data registers. The other functions of units 1 and 2 are the same.

- 10-bit resolution
- Input channels
  Unit 1: 12 channels for the H8S/20223 and H8S/20203 Groups and 8 channels for the H8S/20103 Group
  Unit 2: 4 channels for the H8S/20223 Group
- Conversion time: 2 µs per channel (at 20 MHz operation)
- Operating modes: Two
  A/D conversion mode: A selected analog input is A/D converted.
  Compare mode: A selected analog input is compared with the voltage specified by the user.
- Channel select modes
  Single mode: Single-channel A/D conversion or comparison
  Scan mode: Continuous A/D conversion on 1 to 4 channels, or 1 to 8 channels
- Data registers: 8 data registers for unit 1 and 4 data registers for unit 2
  Conversion results are held in a 16-bit data register for each channel.
- Sample and hold function
- Three kinds of conversion start
  Conversion can be started by software, conversion start trigger by 16-bit timer (timer RC or RD), or external trigger signal.
- Interrupt request
  A/D conversion end interrupt (ADI) request can be generated.
- Compare result change interrupt (CMPI) request can be generated.
- Module standby function can be set.

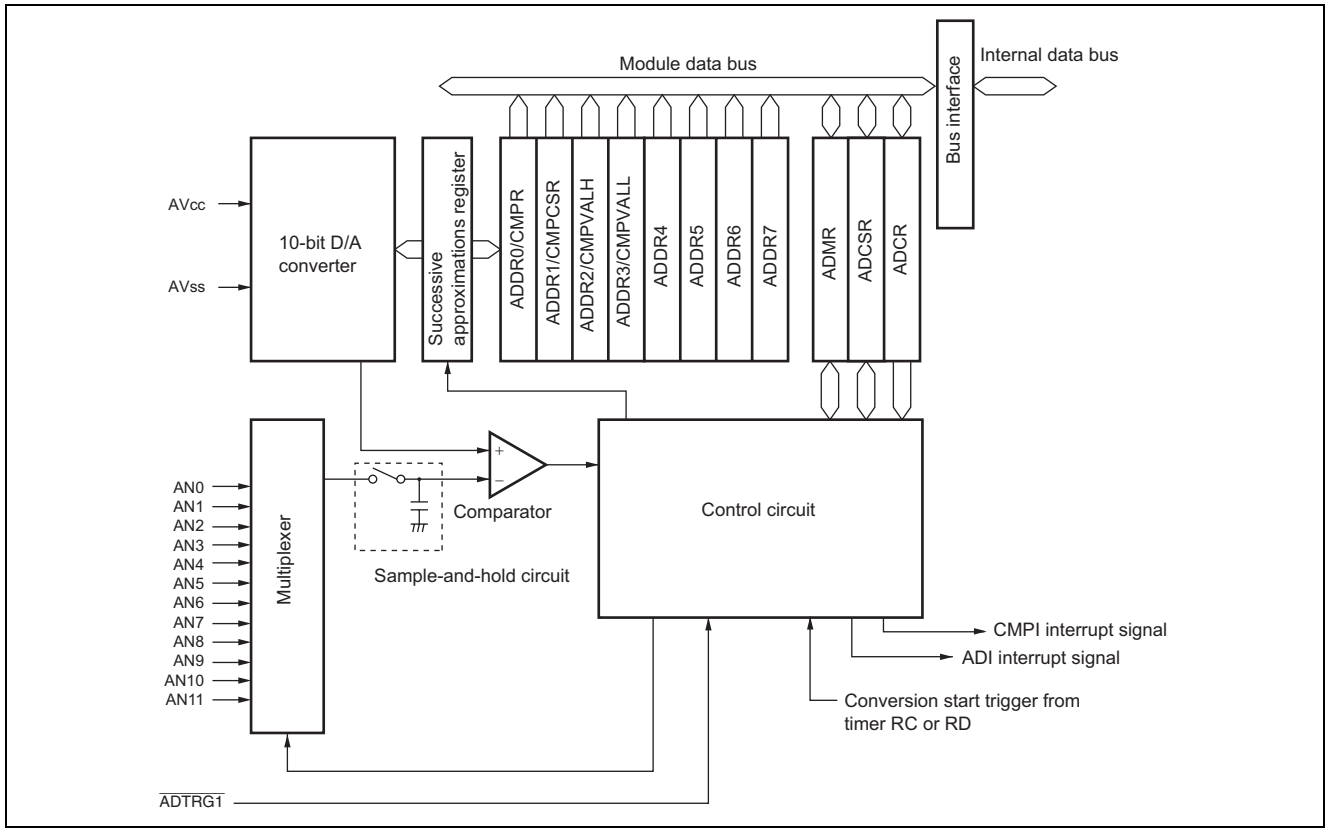


Figure 9 Block Diagram of A/D Converter (Unit 1)

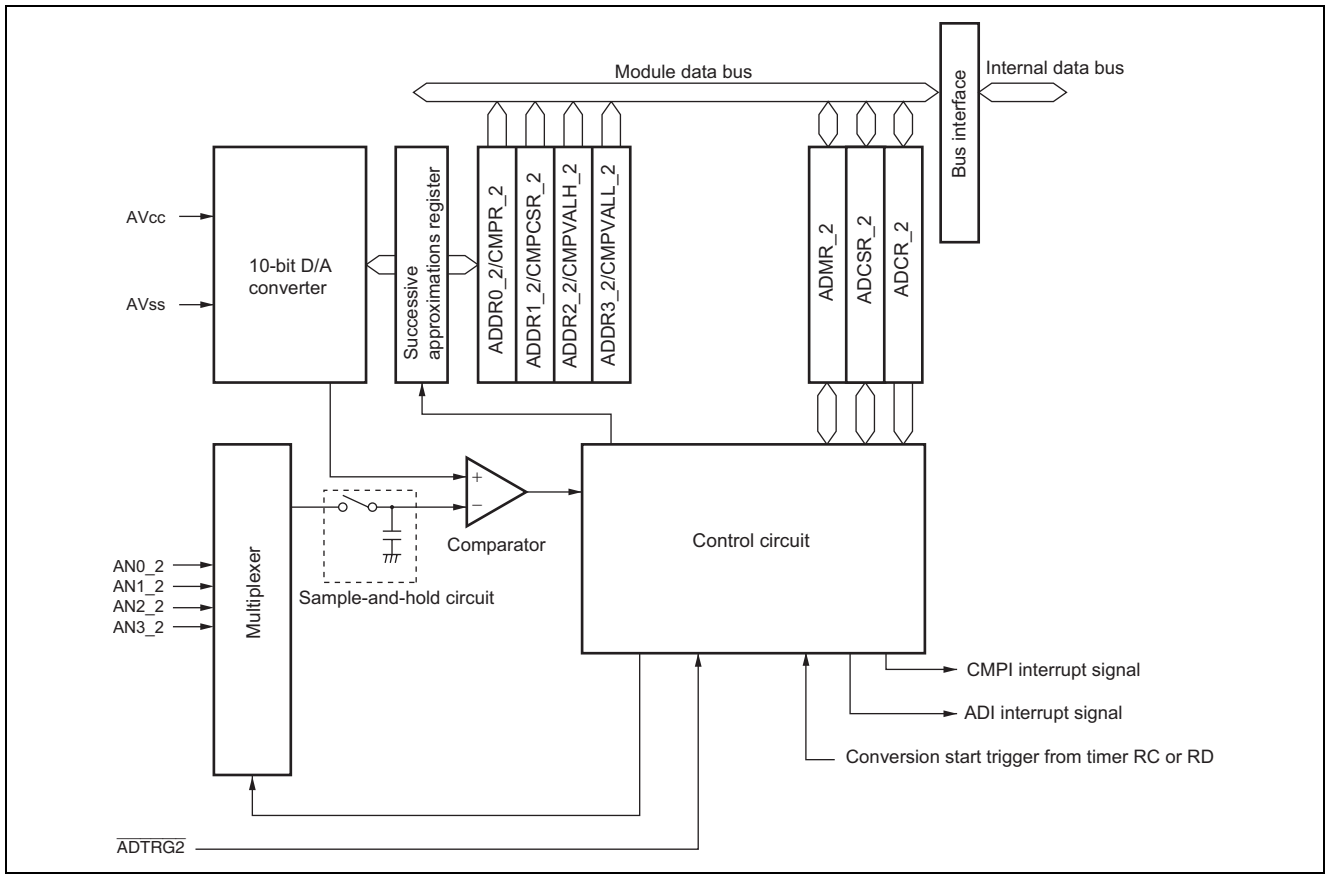
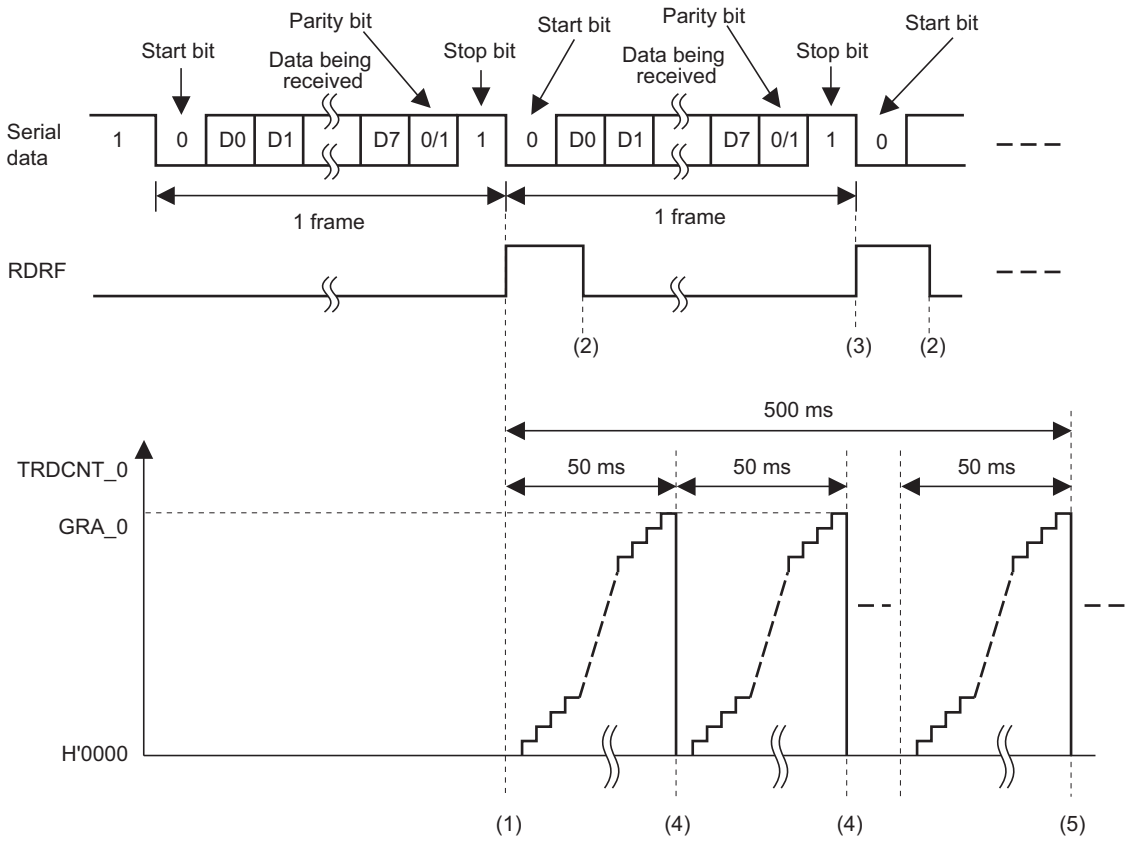


Figure 10 Block Diagram of A/D Converter (Unit 2)

3. Principle of Operation

Figure 11 shows the principle of operation in this sample task. How operation with three events connected by using the ELC is set up and proceeds by means of the hardware and software processing at the numbered points in figure 11 is described in figure 12. Figure 13 shows the timing of operations of the A/D converter in this sample task.

When the RDRF bit is set to 1 and a receive-data-full signal is generated, timer RD_0 channel 0 starts without CPU intervention. When a compare match between timer RD_0 channel 0 and GRA_0 is generated, A/D conversion starts without CPU intervention.



Note: For hardware and software processing from (1) to (5), refer to figure 12.

Figure 11 Principle of Operation in the Sample Task

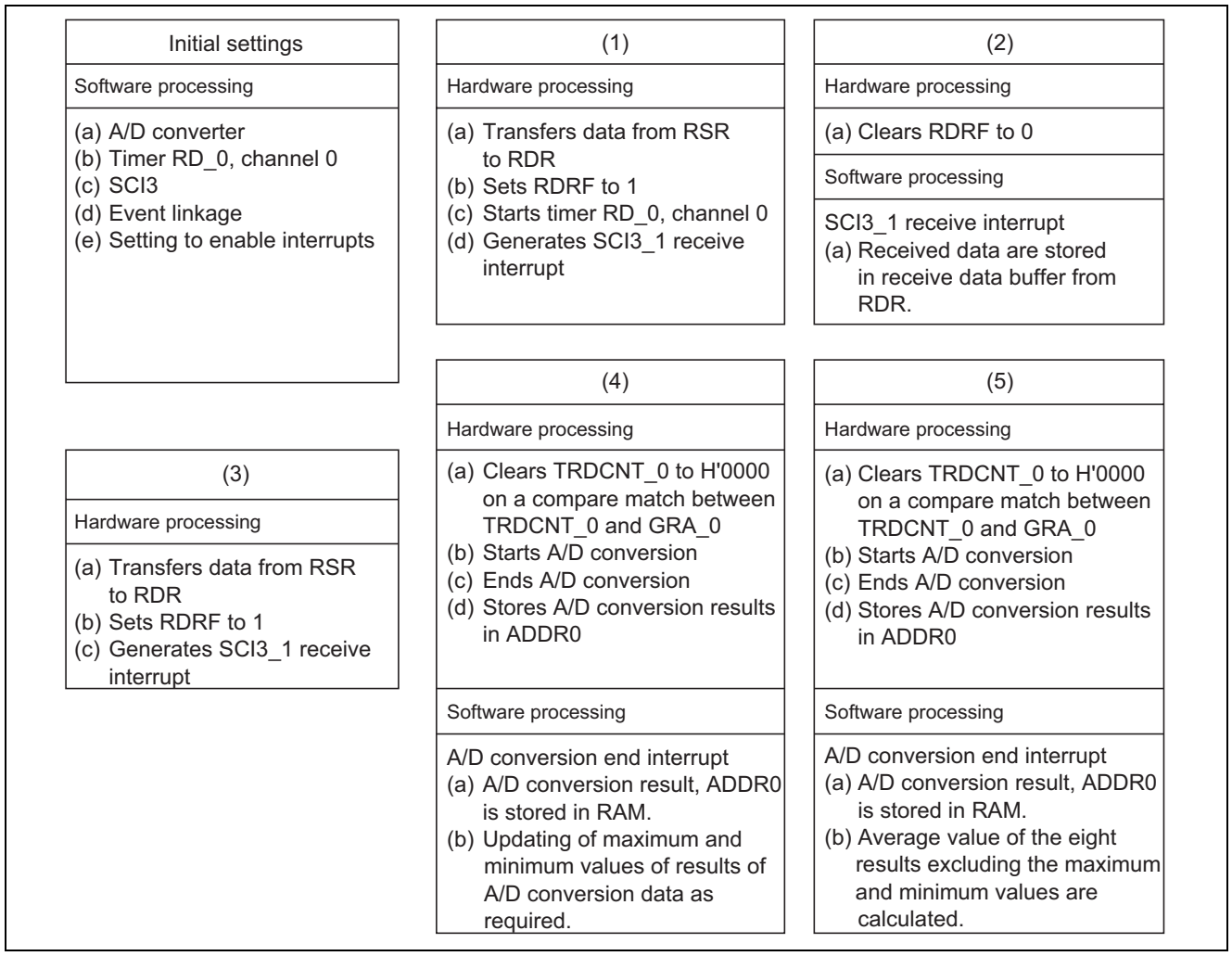


Figure 12 Hardware and Software Processing in the Sample Task

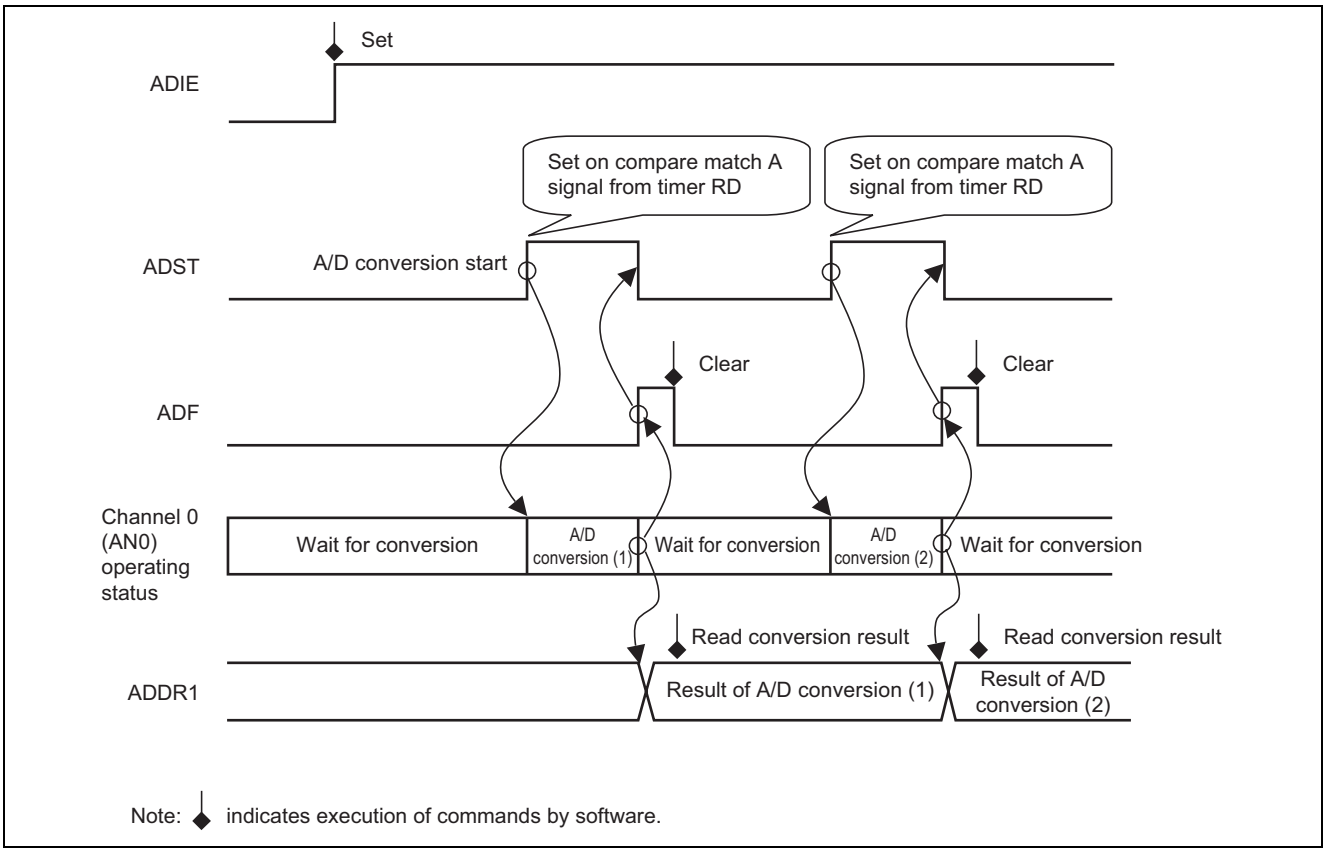


Figure 13 Timing of A/D Converter Operations in the Sample Task

4. Description of Software

4.1 Descriptions of Functions

The functions in this sample task are listed and described in table 4.

Table 4 Description of Functions

Function Name	Label Name	Description
Main routine	main	Calls all functions.
System initialization routine	h8s_sysinit	Makes settings for module standby, system clock and bus-master operating clock, and halts the WDT.
A/D converter setting routine	init_ad	Makes settings for A/D converter and the ELC.
Timer RD setting routine	init_tmrd	Makes settings for timer RD_0 channel 0 and the ELC.
SCI3_1 setting routine	init_sci31	Makes settings for SCI3 channel 1.
A/D conversion-end interrupt routine	INT_IADEND_AD1	Stores results of A/D conversion in RAM and calculates the valid value of the results of A/D conversion.
SCI3_1 reception error interrupt routine	INT_ERI_SCI31	Clears the receive error flag.
SCI3_1 reception interrupt routine	INT_RXI_SCI31	Reads received data and stores the data. Reception is disabled once reception of all bytes is complete.

4.2 Description of Argument

No arguments are used in this sample task.

4.3 Description of Internal Registers

Table 5 gives descriptions of how internal registers are used in this sample task.

Table 5 Description of Internal Registers

Register Name	Symbol	Description	Address	Setting
PMR2	PMR21	The P21 pin is set to operate as the RXD pin.	H'FF0001	1
PMRA	PMRA2	The PB0 pin is set to operate as the AN0 pin.	H'FF0009	1
PMRJ	PMRJ[1:0]	The OSC1 and OSC2 functions are selected for pins PJ0/OSC1 and PJ1/OSC2.	H'FF000C	B'11
PUCR2	PUCR21	The pull-up MOS of the P21 pin is set to ON.	H'FF0011	1
SMR	COM	Asynchronous mode	H'FF0550	0
	CHR	“8 bits” is selected as the data length.		0
	PE	The parity bit is appended in transmission and the parity bit is checked in reception.		1
	PM	Even parity is selected for transmission and reception.		0
	STOP	1 stop bit is selected.		0
	MP	The multiprocessor communications function is disabled.		0
	CKS[1:0]	In conjunction with the BRR setting, this selects 9,600 bps as the bit rate.		B'00
BRR		In conjunction with the setting of the CKS[1:0] bits in SMR, this selects 9,600 bps as the bit rate.	H'FF0551	64
SCR3	TIE	The TXI interrupt request is disabled.	H'FF0552	0
	RIE	RXI and ERI interrupt requests are enabled.		1
	TE	Transmission is disabled.		0
	RE	Reception is enabled.		1
	TEIE	The TEI interrupt request is disabled.		0
	CKE[1:0]	Specifies the on-chip baud rate generator as the clock source.		B'00
SSR	RDRF	[Setting condition] • Reception being completed normally and transfer of received data from RSR to RDR [Clearing conditions] • The CPU writing 0 after reading RDRF as 1 • The CPU reading data from RDR • The DTC transferring data from RDR when the DTC settings satisfying the flag clearing conditions*	H'FF0554	0 or 1
RDR		Received data are read.	H'FF0555	Undefined
ADDR0		Data of A/D conversion 0 are read.	H'FF05E0	Undefined
ADCSR	ADIE	A/D conversion end interrupt is enabled.	H'FF05F0	1
	ADST	A/D conversion is halted.		0
	CH[3:0]	A/D conversion channel is set to AN0.		B'0000

Register Name	Symbol	Function	Address	Setting
ADCR	TRGS[1:0]	In combination with EXTRGS, this disables the initiation of A/D conversion by an external trigger.	H'FF05F1	B'00
	SCANE	Single mode		0
	CKS[1:0]	A/D conversion time = 84 states		B'10
	EXTRGS	In combination with the TRGS[1:0] bits, this disables the initiation of A/D conversion by an external trigger.		0
ADMR	ADM1	A/D conversion mode	H'FF05F4	0
ELSR3		Setting is made to link timer RD_0 channel 0 and a receive-data-full signal from SCI3_1.	H'FF0683	H'4C
ELSR10		Setting is made to link A/D conversion and compare match signal A from timer RD_0 channel 0.	H'FF068A	H'09
ELOPA	TMRD1M[2:1]	These bits are set so that the counter of timer RD_0 channel 0 starts counting at the time of event-signal input.	H'FF06B5	B'00
ELCR	ELCON	Linkage is enabled for all events .	H'FF06BC	1
SYSCCR	PHIHSEL	ϕosc is selected for clock source ϕhigh.	H'FF06D0	1
LPCR1	PSCSTP	PSC divider is operating.	H'FF06D1	0
	PHIBSEL	ϕhigh is selected for base clock source ϕ.		1
LPCR2	PHI[2:0]	ϕbase is selected for system clock ϕ.	H'FF06D2	B'000
LPCR3	PHIS[2:0]	ϕ is selected for bus master operation clock ϕs.	H'FF06D3	B'000
OSCCSR		Setting is made for ϕosc oscillation settling time.	H'FF06D5	H'0E
TMWD		Clock input to the WDT is prohibited.	H'FFFF99	H'F7
TCSRWD		Writing to the TMWD register is controlled.	H'FFFF9A	H'A3
TRDCNT_0		TRDCNT_0 is initialized.	H'FFFFB0	H'0000
GRA_0		Setting is made for the period of timer RD_0, channel 0 (period of A/D conversion).	H'FFFFB2	H'7A11
TRDCR	CCLR[2:0]	TRDCNT is cleared by GRA compare match/input capture.	H'FFFFC4	B'001
	CKEG[1:0]	Counting of rising edges		B'00
	TPSC[2:0]	Internal clock: count by ϕ/32		B'100
TRDSTR	CSTPN0	Counting proceeds on a compare match of TRDCNT_0 and GRA_0.	H'FFFFD2	1
	STR0	Counting by TRDCNT_0 stops.		0
MSTCR1	MSTWDT	Watchdog timer is released from module standby.	H'FFFFDC	0
	MSTAD1	A/D converter unit 1 is released from module standby.		0
MSTCR2	MSTSCI3_1	SCI3 channel 1 is released from module standby.	H'FFFFDD	0
MSTCR3	MSTTMRD1	Timer RD unit 0 is released from module standby.	H'FFFFDE	0

Note: * The DTC clears the peripheral module flags when all of the following three conditions are satisfied:
1. The DISEL bit is 0.
2. The value in the transfer counter (count register CRA in normal and repeat modes or count register CRB in block mode) is not 0.
3. A chain transfer is not used.

4.4 RAM Usage

Table 6 gives a description of RAM usage in this sample task.

Table 6 RAM Usage

Label Name	Description	Memory	Name of Employing Module
ad_sum	Holds the total of results of A/D conversion	4 bytes	init_ad, AINT_IADEND_AD1
ad_data[]	Holds results of A/D conversion	2 bytes	INT_IADEND_AD1
ad_fix	Holds valid data (average of values other than max. and min.) from A/D conversion	2 bytes	INT_IADEND_AD1
ad_max	Holds maximum value of A/D conversion results	2 bytes	INT_IADEND_AD1
ad_min	Holds minimum value of A/D conversion results	2 bytes	INT_IADEND_AD1
rcv_buf[]	SCI3_1 received data buffer	1 byte × 8	INT_RXI_SCI31
index_ad	Index for data produced by A/D conversion	1 byte	init_ad, AINT_IADEND_AD1
rcv_index	SCI3_1 received data buffer index	1 byte	init_sci31, AINT_RXI_SCI31

4.5 Description of Definitions in Use

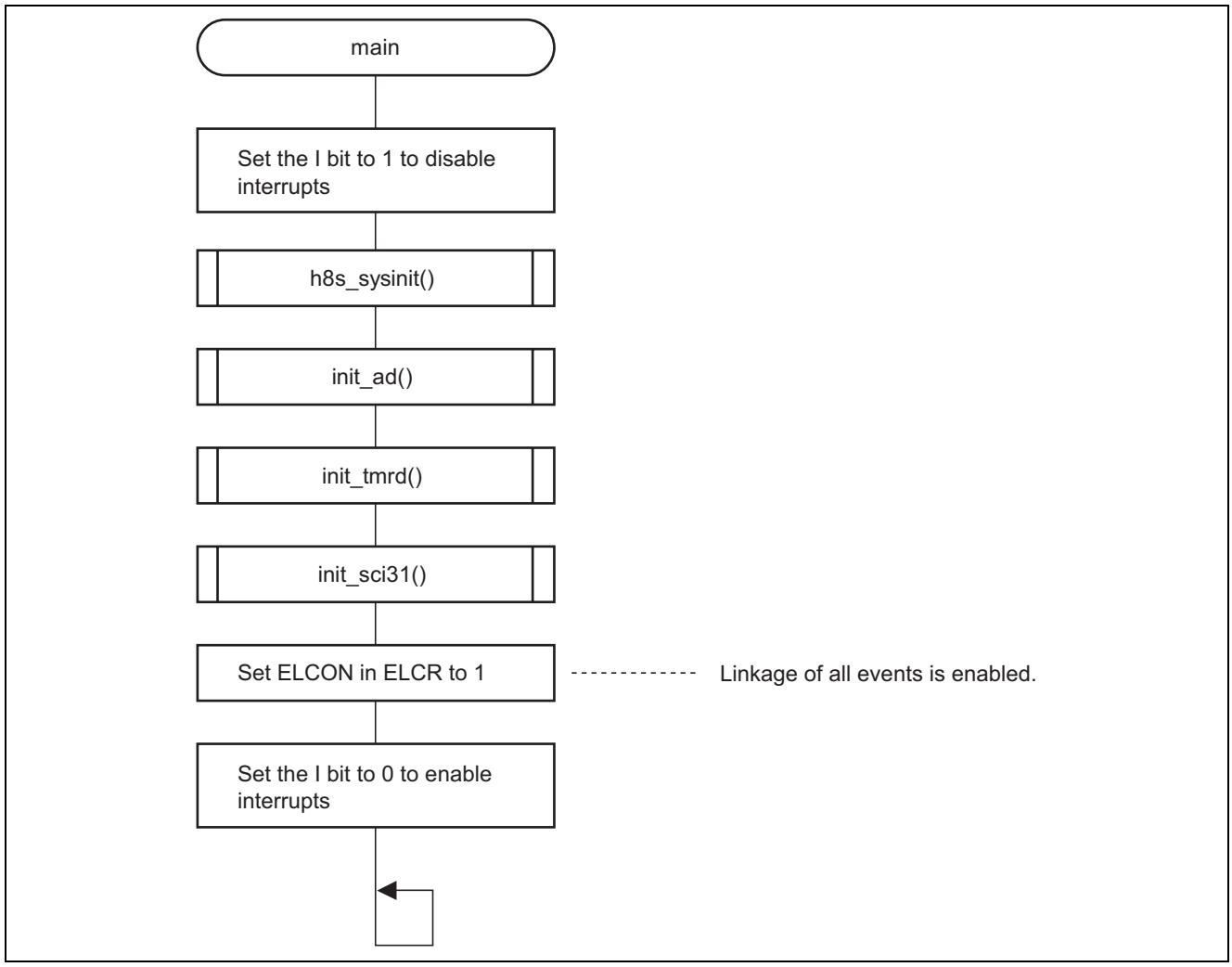
Table 7 gives description of the definitions used in this sample task.

Table 7 Description of Definitions in Use

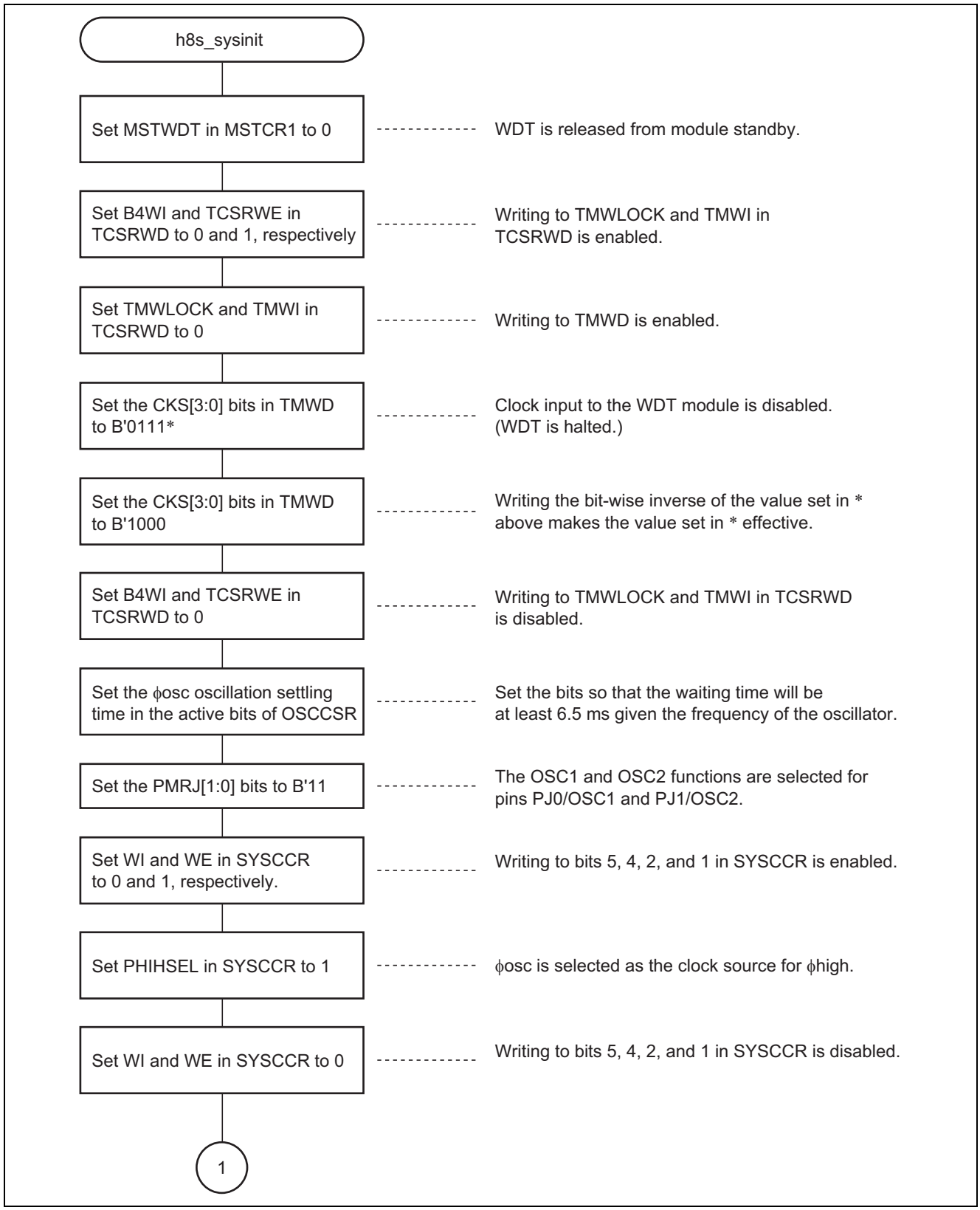
Label Name	Description	Constant
AD_CYCLE	Timer RD ϕ = 20 MHz, counting of ϕ/32, generated value = 50 ms	H'7A11
SAMPLING	Number of times sampling proceeds for overall A/D conversion (10-1)	9
FRAME_SIZE	Specification of size for frame reception	8

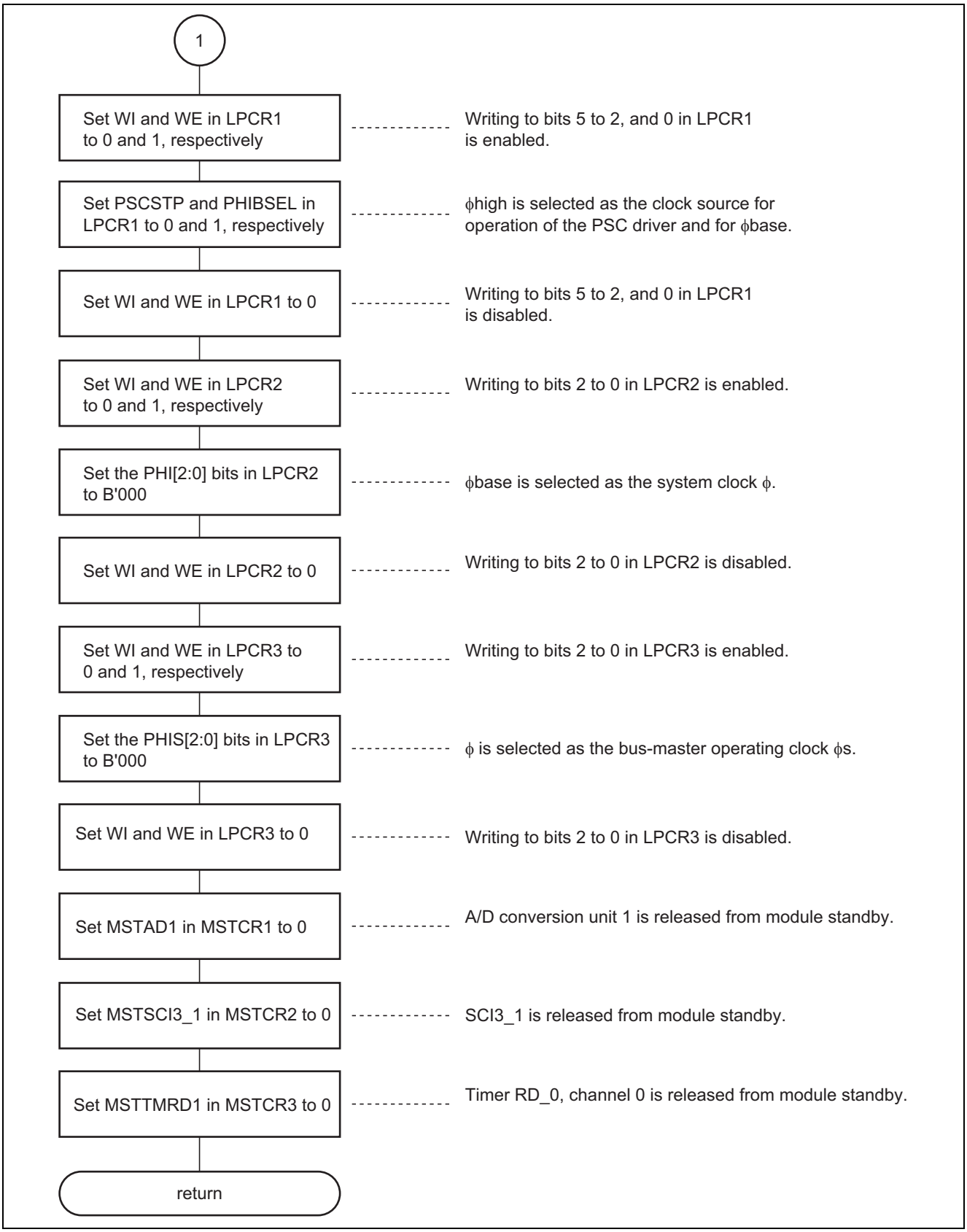
5. Flowcharts

5.1 Main Routine

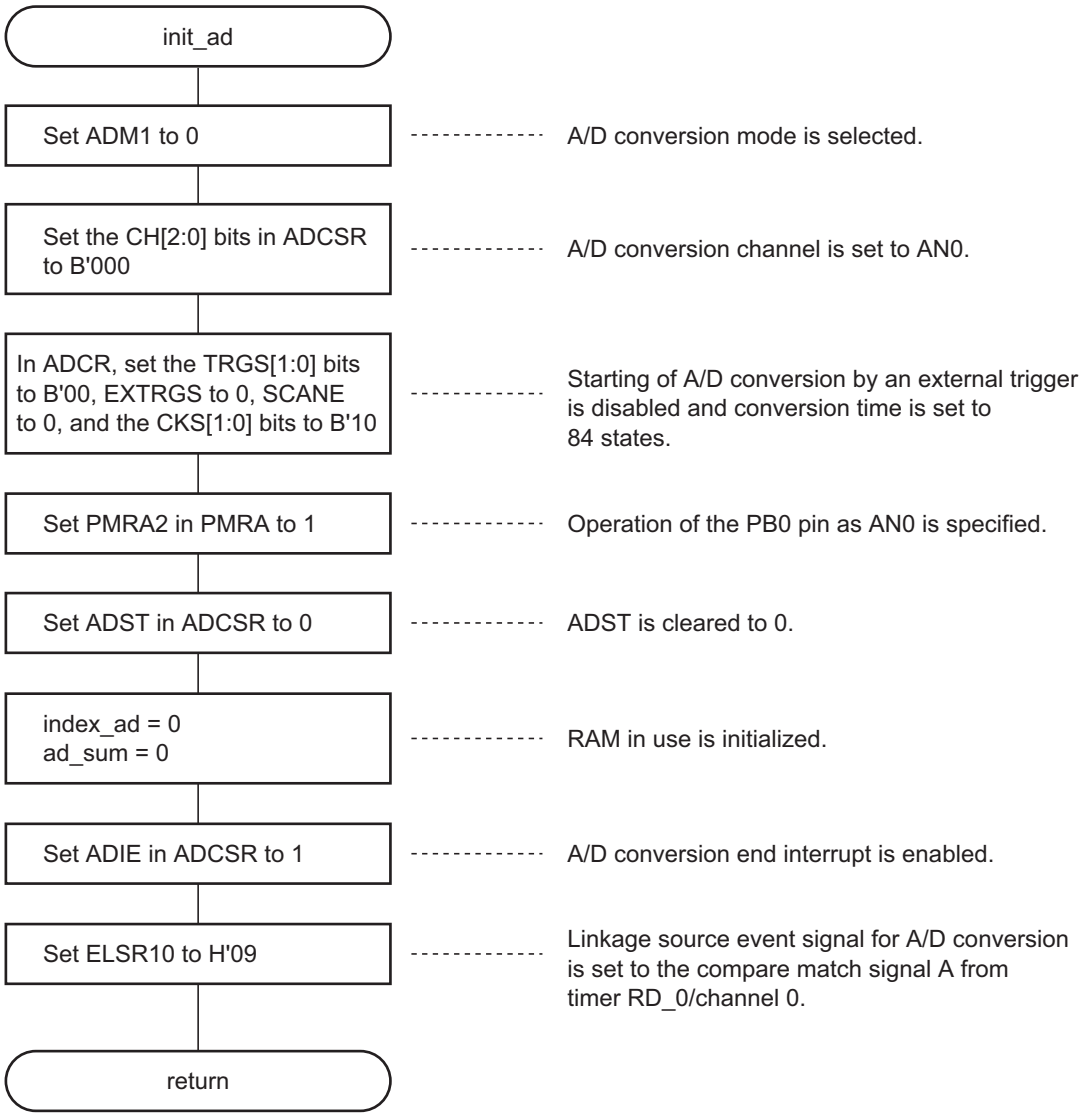


5.2 System Initialization Routine

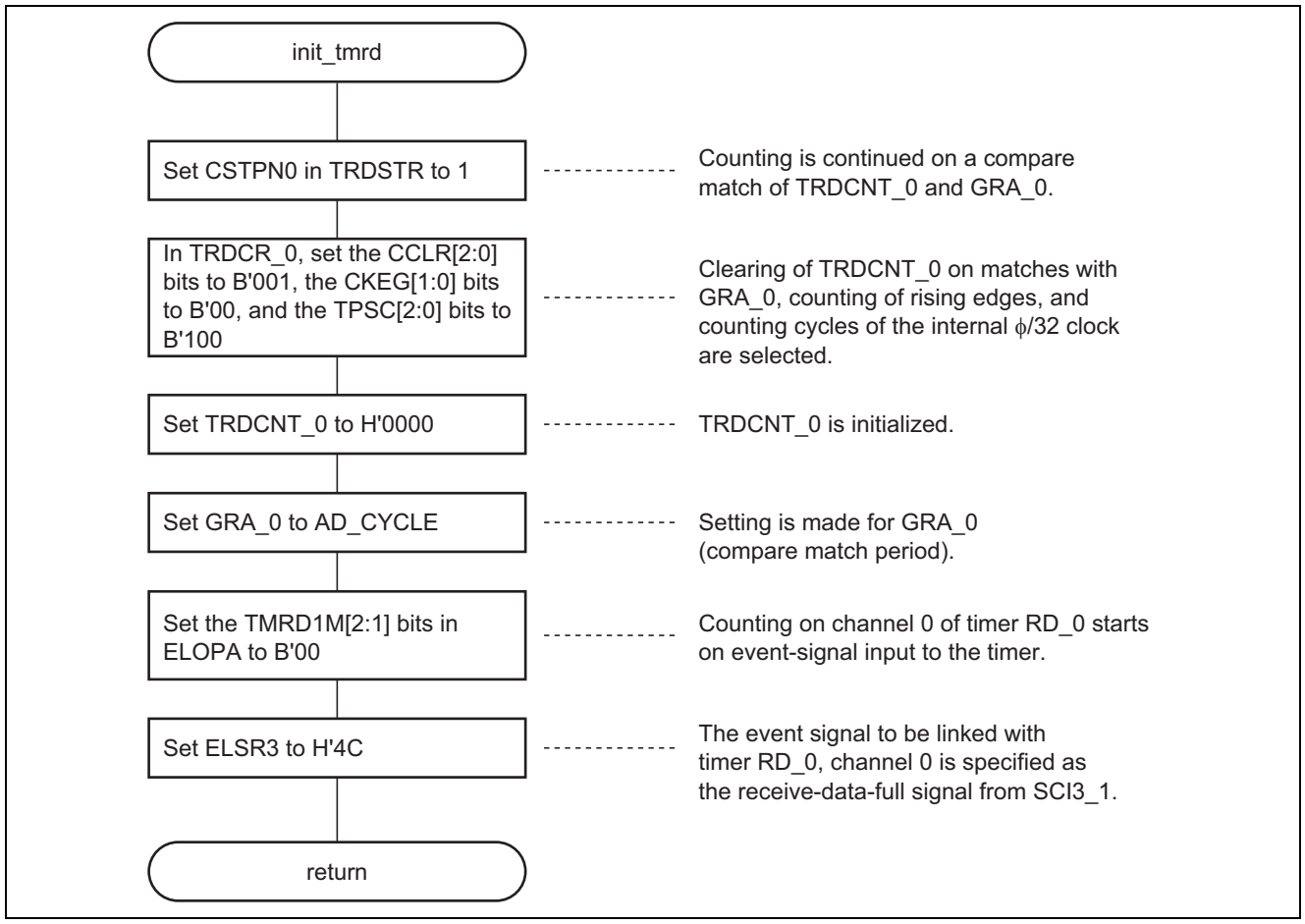



1
Set WI and WE in LPCR1 to 0 and 1, respectively
Writing to bits 5 to 2, and 0 in LPCR1 is enabled.
Set PSCSTP and PHIBSEL in LPCR1 to 0 and 1, respectively
ϕhigh is selected as the clock source for operation of the PSC driver and for ϕbase.
Set WI and WE in LPCR1 to 0
Writing to bits 5 to 2, and 0 in LPCR1 is disabled.
Set WI and WE in LPCR2 to 0 and 1, respectively
Writing to bits 2 to 0 in LPCR2 is enabled.
Set the PHI[2:0] bits in LPCR2 to B'000
ϕbase is selected as the system clock ϕ.
Set WI and WE in LPCR2 to 0
Writing to bits 2 to 0 in LPCR2 is disabled.
Set WI and WE in LPCR3 to 0 and 1, respectively
Writing to bits 2 to 0 in LPCR3 is enabled.
Set the PHIS[2:0] bits in LPCR3 to B'000
ϕ is selected as the bus-master operating clock ϕs.
Set WI and WE in LPCR3 to 0
Writing to bits 2 to 0 in LPCR3 is disabled.
Set MSTAD1 in MSTCR1 to 0
A/D conversion unit 1 is released from module standby.
Set MSTSCI3_1 in MSTCR2 to 0
SCI3_1 is released from module standby.
Set MSTTMRD1 in MSTCR3 to 0
Timer RD_0, channel 0 is released from module standby.
return

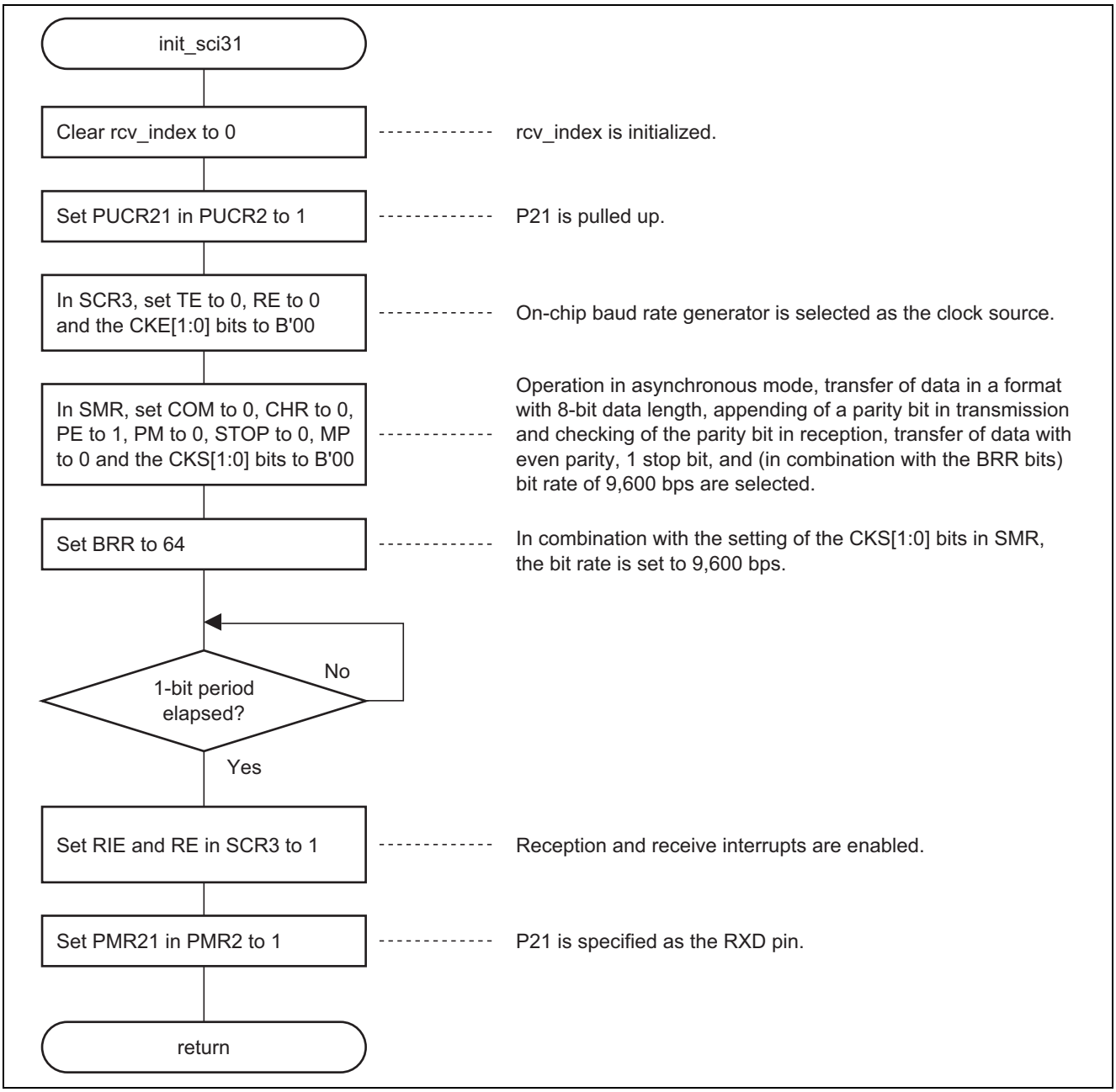
5.3 A/D Converter Setting Routine



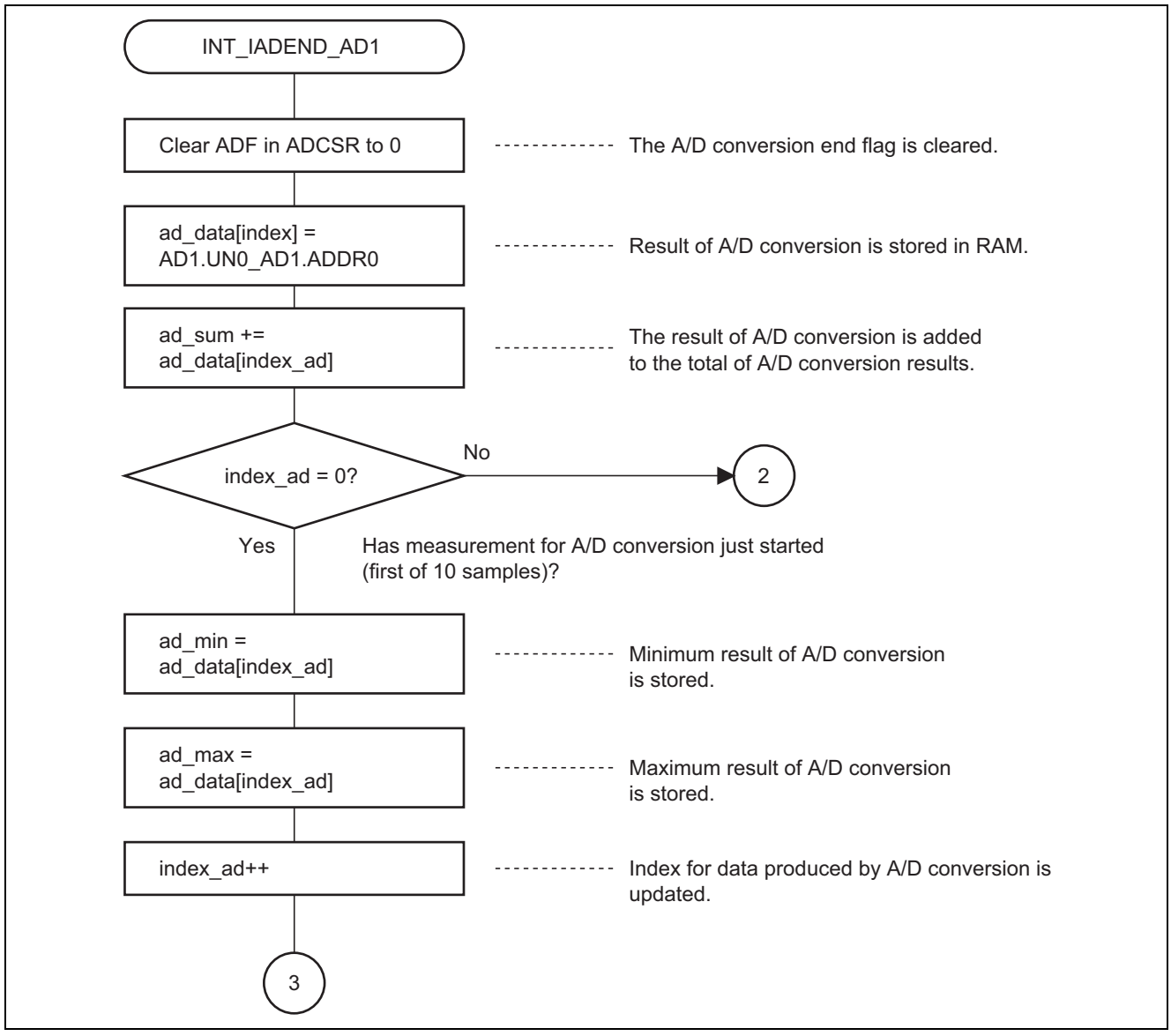
5.4 Timer RD Setting Routine

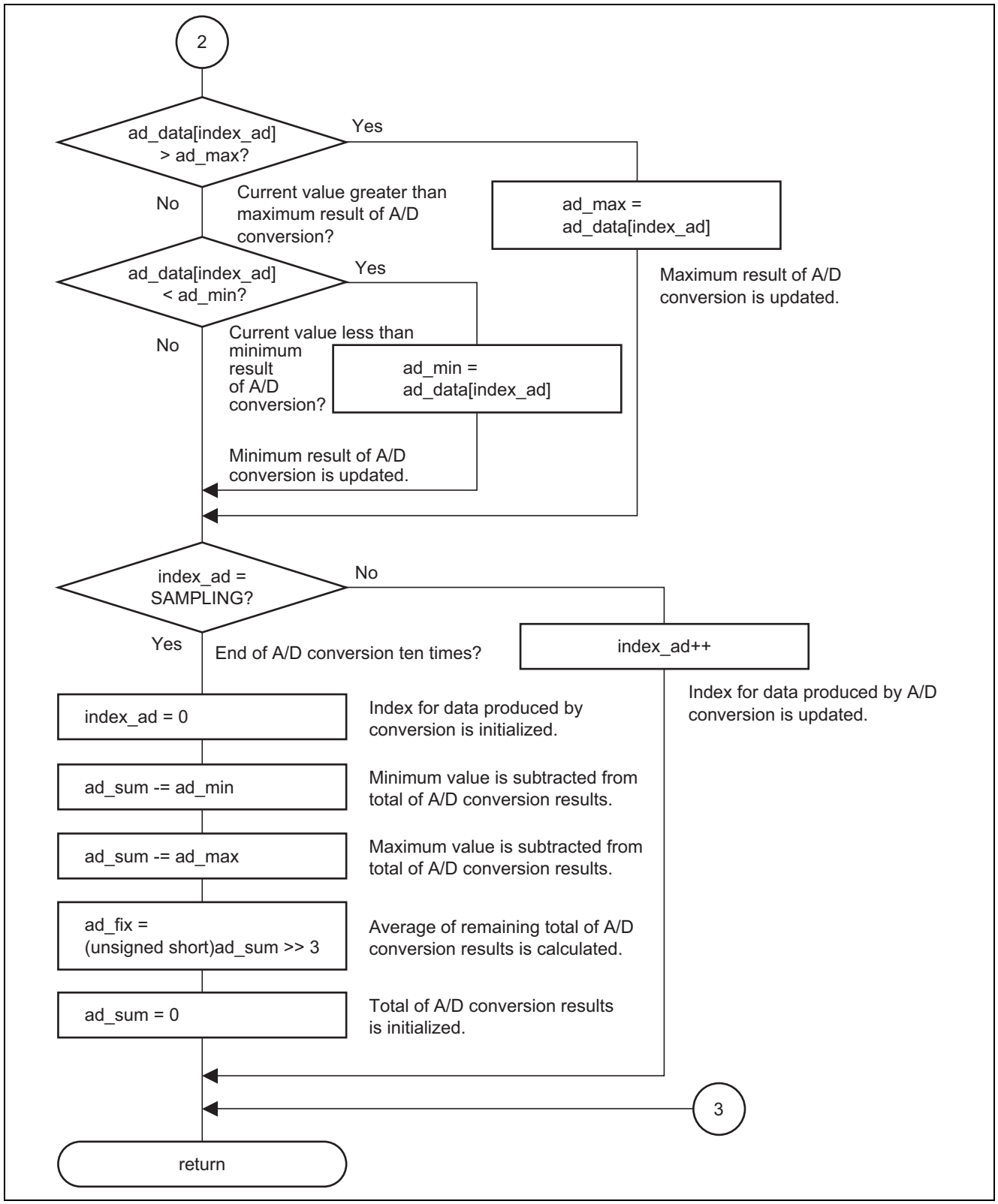


5.5 SCI3_1 Setting Routine

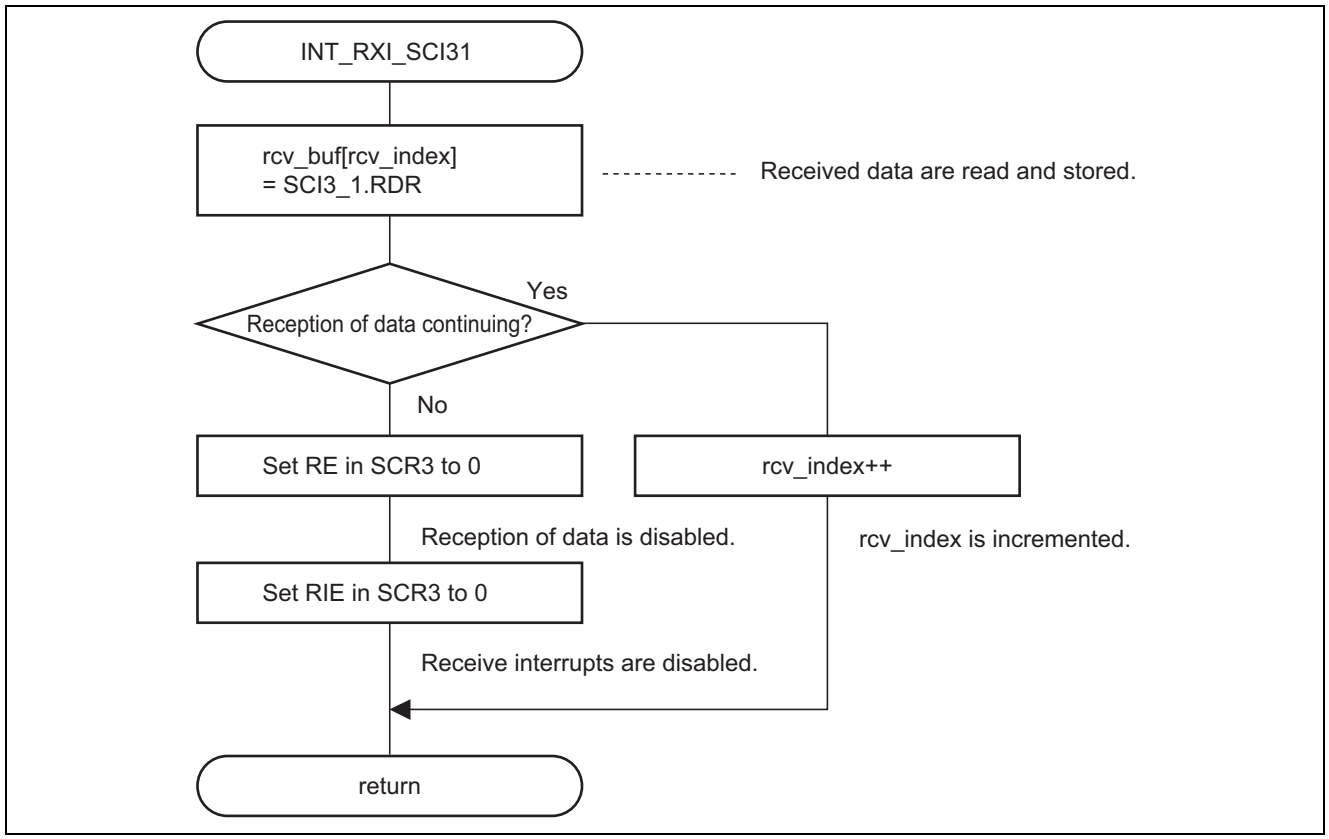


5.6 A/D Conversion End Interrupt Routine

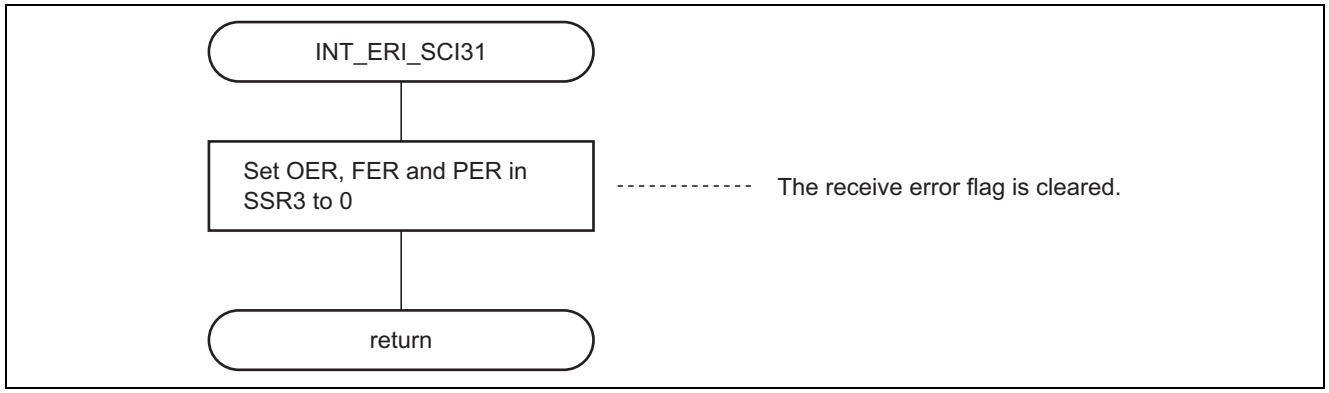



2
ad_data[index_ad] > ad_max?
Yes
No
Current value greater than maximum result of A/D conversion?
ad_max = ad_data[index_ad]
Maximum result of A/D conversion is updated.
ad_data[index_ad] < ad_min?
Yes
No
Current value less than minimum result of A/D conversion?
ad_min = ad_data[index_ad]
Minimum result of A/D conversion is updated.
index_ad = SAMPLING?
No
Yes
End of A/D conversion ten times?
index_ad++
Index for data produced by A/D conversion is updated.
index_ad = 0
Index for data produced by conversion is initialized.
ad_sum -= ad_min
Minimum value is subtracted from total of A/D conversion results.
ad_sum -= ad_max
Maximum value is subtracted from total of A/D conversion results.
ad_fix = (unsigned short)ad_sum >> 3
Average of remaining total of A/D conversion results is calculated.
ad_sum = 0
Total of A/D conversion results is initialized.
3
return

5.7 SCI3_1 Receive Interrupt Routine



5.8 SCI3_1 Receive Error Interrupt Routine



6. Program Listing

```
<sci3_trd_adc.c>
/************************************************************************/
/* H8S/2000 Tiny Series -H8S/20203-                                     */
/* Application Note                                                     */
/*                                                                      */
/* 3 event combination by ELC                                           */
/*                                                                      */
/* Function                                                             */
/* : 3 event combination by ELC (start A/D converter by SCI3_1          */
/* receive data full, Timer RD unit0 channel 0 compare match A)         */
/* Event Link A/D converter, Timer RD and SCI3_1                        */
/*                                                                      */
/* External Clock : 20 MHz                                              */
/* Internal Clock : 20 MHz                                              */
/************************************************************************/
#include <machine.h>
#include "iodefine.h"

/* Timer RD */
/* unit0 (Phi=20MHz, divide 32) */
#define AD_CYCLE          0x7A11         /* Set GRA_0 (50 ms) */

#define SAMPLING          9              /* Set A/D sampling (SAMPLING+1) */

/* SCI3_1 */
#define FRAME_SIZE        8              /* frame size */

/******************************************************/
/* Declaration of function prototypes                 */
/******************************************************/
void main(void);
void init_ad(void);
void init_tmrd(void);
void init_sci31(void);
void h8s_sysinit(void);
```

```
/*****************************************************/
/* Definition of RAM area                            */
/*****************************************************/
unsigned long ad_sum;                   /* A/D sum data */
unsigned short ad_data[SAMPLING+1];     /* save A/D data AN0 */
unsigned short ad_fix;                  /* save A/D fix data */
unsigned short ad_max;                  /* save maximum A/D data AN0 */
unsigned short ad_min;                  /* save minimum A/D data AN0 */
unsigned char index_ad;                 /* index of A/D data buffer */
unsigned char rcv_buf[FRAME_SIZE];      /* reception buffer */
unsigned char rcv_index;                /* reception index */

/*****************************************************/
/* Name: main                                        */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: User main                            */
/*****************************************************/
void main(void)
{
    set_ccr(0x80);                          /* set CCR-Ibit */

    h8s_sysinit();                          /* initialize system */

    init_ad();                              /* initialize A/D converter */

    init_tmrd();                            /* initialize timer RD */

    init_sci31();                           /* initialize SCI3_1 */

    set_imask_ccr(0);                       /* interrupt enable */

    ELC.ELCR.BIT.ELCON = 1;                 /* event link enable */

    while(1);
}
```

```
/*****************************************************/
/* Name: init_sci31                                  */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: initialize SCI3_1                    */
/*****************************************************/
void init_sci31(void)
{

    unsigned short wait;

    rcv_index = 0;                /* clear rcv_index */

    /* port pull up */
    PUCR2.BYTE |= 0x02;           /* pull up P21 */

    SCI3_1.SCR3.BYTE = 0x00;      /* clear TE, RE */
                                  /* internal baud rate generator */
    SCI3_1.SMR.BYTE = 0x20;       /* select asynchronous mode */
                                  /* even parity, 1 stop bit */
    SCI3_1.BRR = 64;              /* bit rate => 9600 bit/s */

    /* 1bit wait */
    for( wait=0; wait<420; wait++ );

    SCI3_1.SCR3.BYTE |= 0x50;     /* set RIE, RE */
    PMR2.BIT.PMR21 = 1;           /* P21=>RXD */
}

/*****************************************************/
/* Name: init_tmrd                                   */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: initialize timer RD                  */
/*****************************************************/
void init_tmrd(void)
{
    /* unit 0 */
    TRD01.TRDSTR.BYTE = 0xFC;     /* continue count when TRDCNT_0 compare match GRA_0, TRDCNT_0 stop */
```

```
    TRD0.TRDCR.BYTE = 0x24;           /* clear TRDCNT_0 when compare match GRA_0 */
                                      /* select clock Phi/32 */
    TRD0.TRDCNT = 0x0000;             /* clear TRDCNT_0 */
    TRD0.GRA = AD_CYCLE;              /* set A/D convert cycle */

    /* Set event link, SCI3_1 receive data full */
    ELC.ELOPA.BYTE = 0xFC;            /* Timer RD_0 count start when event input */
    ELC.ELSR3.BYTE = 0x4C;
}

/*****************************************************/
/* Name: init_ad                                     */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: initialize A/D                       */
/*****************************************************/
void init_ad(void)
{

    AD1.ADMR.BYTE = 0x00;             /* select A/D converter mode */
    AD1.ADCSR.BYTE = 0x00;            /* channel select AN0 */
    AD1.ADCR.BYTE = 0x08;             /* single mode CH1 */
                                      /* A/D conversion time => 84 states */
    PMRA.BIT.PMRA2 = 1;               /* PB0 => AN0 */

    AD1.ADCSR.BIT.ADST = 0;           /* clear ADST */

    index_ad = 0;                     /* clear index_ad */
    ad_sum = 0;                       /* clear ad_sum */

    AD1.ADCSR.BIT.ADIE = 1;           /* A/D interrupt enable */

    /* Set event link, Timer RD unit0 channel 0 compare match A */
    ELC.ELSR10.BYTE = 0x09;
}
```

```
/*****************************************************/
/* Name: h8s_sysinit                                 */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: initialize H8S/20203                 */
/*****************************************************/
void h8s_sysinit(void)
{
    MSTCR1.BIT.MSTWDT = 0;                              /* WDT module standby off */

    /* stop WDT */
    WDT.TCSRWD.BYTE = 0x97;                             /* write enable TMWLOCK, TMWI */
    WDT.TCSRWD.BYTE = 0xA3;                             /* write enable TMWD */
    WDT.TMWD.BYTE = 0xF7;                               /* Not select clock source */
    WDT.TMWD.BYTE = 0xF8;                               /* write bit inversion */
    WDT.TCSRWD.BYTE = 0x87;                             /* write disable TMWLOCK, TMWI */

    CPG.OSCCSR.BYTE = 0x0E;                             /* wait for 6.5 ms, Phi_osc = 20 MHz */
    PMRJ.BYTE = 0x03;                                   /* select OSC1,OSC2 */

    CPG.SYSCCR.BYTE = (CPG.SYSCCR.BYTE & 0x7F)| 0x40;   /* WI=0, WE=1 */
    CPG.SYSCCR.BYTE = 0x60;                             /* high = Phi_osc, Phi_low = Phi_loco */
    CPG.SYSCCR.BYTE = CPG.SYSCCR.BYTE & 0x3F;           /* WI = 0, WE = 0 */

    CPG.LPCR1.BYTE = (CPG.LPCR1.BYTE & 0x7F) | 0x40;    /* WI = 0, WE = 1 */
    CPG.LPCR1.BYTE = 0x41;                              /* PSC on, Phi_base = Phi_high */
    CPG.LPCR1.BYTE = CPG.LPCR1.BYTE & 0x3F;             /* WI = 0, WE = 0 */

    CPG.LPCR2.BYTE = (CPG.LPCR2.BYTE & 0x7F) | 0x40;    /* WI = 0, WE = 1 */
    CPG.LPCR2.BYTE = 0x40;                              /* select system clock */
    CPG.LPCR2.BYTE = CPG.LPCR2.BYTE & 0x3F;             /* WI = 0, WE = 0 */

    CPG.LPCR3.BYTE = (CPG.LPCR3.BYTE & 0x7F) | 0x40;    /* WI = 0, WE = 1 */
    CPG.LPCR3.BYTE = 0x40;                              /* select clock of bus master */
    CPG.LPCR3.BYTE = CPG.LPCR3.BYTE & 0x3F;             /* WI = 0, WE = 0 */

    /* module standby off */
    MSTCR1.BIT.MSTAD1 = 0;                              /* A/D module standby off */
    MSTCR2.BIT.MSTSCI3_1 = 0;                           /* SCI3_1 module standby off */
    MSTCR3.BIT.MSTTMRD1 = 0;                            /* Timer RD unit0 module standby off */
}
```

```
<intprg.c>
/******************************************************/
/* Definition of const data                           */
/******************************************************/
#define SAMPLING          9                    /* Set A/D sampling (SAMPLING+1) */

/* SCI3_1 */
#define FRAME_SIZE        8                    /* frame size */

/******************************************************/
/* Extern Definition of RAM area                      */
/******************************************************/
extern unsigned long ad_sum;                   /* A/D sum data */
extern unsigned short ad_data[SAMPLING+1];     /* save A/D data AN0 */
extern unsigned short ad_fix;                  /* save A/D fix data */
extern unsigned short ad_max;                  /* save maximum A/D data AN0 */
extern unsigned short ad_min;                  /* save minimum A/D data AN0 */
extern unsigned char index_ad;                 /* index of A/D data buffer */
extern unsigned char rcv_buf[FRAME_SIZE];      /* reception buffer */
extern unsigned char rcv_index;                /* reception index */

// vector 30 END AD1
__interrupt(vect=30) void INT_IADEND_AD1(void) {
    AD1.ADCSR.BIT.ADF = 0;                     /* clear ADF */

    ad_data[index_ad] = AD1.UN0_AD1.ADDR0;     /* read ADDR0 */
    ad_sum += ad_data[index_ad];               /* add to ad_sum */

    /* start sampling ? */
    if( index_ad == 0 ){
            ad_min = ad_data[index_ad];        /* update ad_min */
            ad_max = ad_data[index_ad];        /* update ad_max */
            index_ad++;
    }
    else{
            /* ad_max over ? */
            if ( ad_data[index_ad] > ad_max ){
                    ad_max = ad_data[index_ad];     /* update ad_max */
            }
            /* ad_min under ? */
            else if( ad_data[index_ad] < ad_min ){
                    ad_min = ad_data[index_ad];     /* update ad_min */
            }

            /* sampling end ? */
            if( index_ad == SAMPLING ){
```

```
                    index_ad = 0;
                    ad_sum -= ad_min;
                    ad_sum -= ad_max;

                    /* calculate average ad_data */
                    ad_fix = (unsigned short)(ad_sum >> 3);
                    ad_sum = 0;
            }
            else{
                    index_ad++;
            }
    }
}

// vector 37 ERI SCI31
__interrupt(vect=37) void INT_ERI_SCI31(void) {
    /* clear receive error flag */
    SCI3_1.SSR.BYTE &= ~0x38;
}

// vector 38 RXI SCI31
__interrupt(vect=38) void INT_RXI_SCI31(void) {
    /* read received data */
    rcv_buf[rcv_index] = SCI3_1.RDR;

    /* continue reception */
    if ( rcv_index < (FRAME_SIZE-1) ){
            rcv_index++;
    }
    /* frame end */
    else{
            SCI3_1.SCR3.BIT.RE = 0;        /* disable reception */
            SCI3_1.SCR3.BIT.RIE = 0;       /* disable receive data full interrupt */
    }
}
```

6.1 Designation of Linkage Addresses

Section Name	Address
PResetPRG, PIntPRG	H'000400
P, C$DSEC, C$BSEC, D	H'000800
B, R	H'FFDF80
S	H'FFFD80

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Dec.24.08	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.