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 Transfer Frames through the SCI3 Module

Introduction

The event link controller (ELC) embedded in products of the H8S/20103, H8S/20203, and H8S/20223 Groups is used to link the data transfer controller (DTC) with frame transfer through SCI3, realizing the transfer and reception of frames 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 a diagram of transmitting a frame of SCI3_1, figure 2 shows a diagram of receiving a frame of SCI3_1, and figure 3 shows a schematic view of transferring a frame.

1. A table of data for transmission through SCI3_1 is created and placed to ROM.
2. The address where the DTC register information starts is stored in the address corresponding to the activation source within the DTC vector table.
3. The transfer of frames is 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.
4. The DTC is placed in normal mode and the size of blocks for transfer is set to one byte.
5. The source address for data transfer is specified as the address of the table and the destination address is specified as the address of TDR in SCI3_1.
6. The event selected in ELSR30 is set as the activation source for the DTC.
7. Transmit-data-empty signal from SCI3_1 is selected in ELSR30 as the event signal to be linked with DTC operation.
8. The event interrupt corresponding to ELSR30 is enabled.
9. Event linkage is enabled.
10. The first byte in the table of data for transmission by SCI3_1 is transmitted by software.
11. The I bit is cleared to enable interrupts.
12. Every time the transmission data empty signal from SCI3_1 is generated, the DTC is activated without CPU intervention and writes data for transmission to TDR of SCI3_1, which proceeds with consecutive transmission.
13. When all 32 bytes from the table have been transmitted, transmission is disabled and operation is switched to frame reception.
14. The source address for data transfer is specified as the address of RDR in SCI3_1 and the transfer destination address is specified as the address where the reception buffer starts.
15. The receive-data-full signal from SCI3_1 is selected as the event signal linked with DTC activation.
16. Every time a receive-data-full signal from SCI3_1 is generated, the DTC is activated without CPU intervention and stores received data in the reception buffer; SCI3_1 proceeds with consecutive reception.
17. Reception is disabled when all 32 bytes have been received.

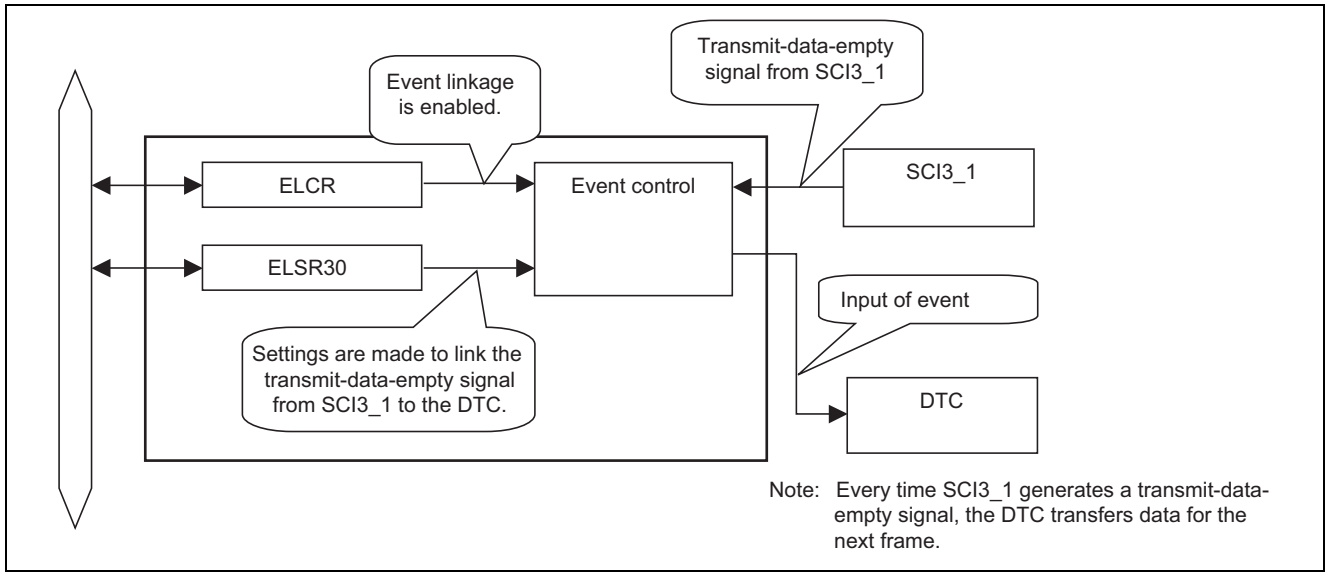


Figure 1 Overview of Using the ELC to Set up Frame Transmission through SCI3_1

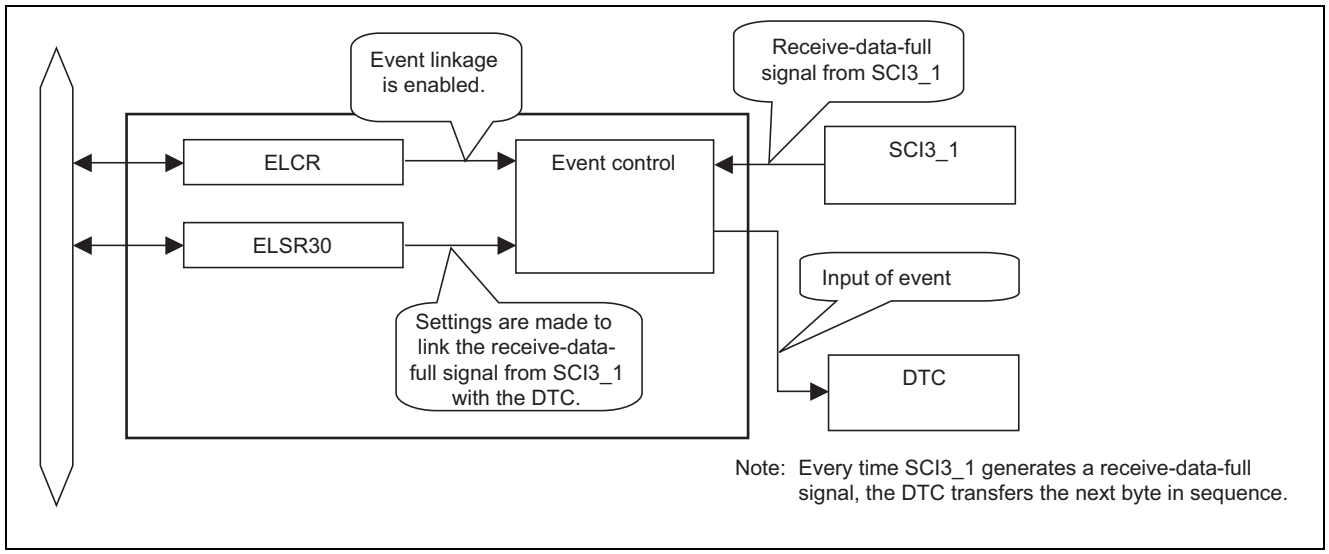


Figure 2 Overview of Using the ELC to Set up Frame Reception through SCI3_1

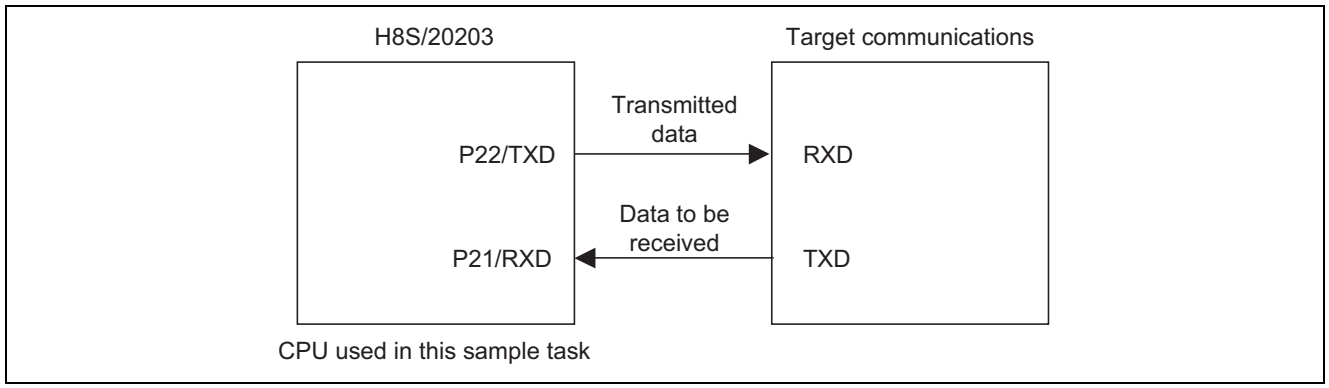


Figure 3 Schematic View of ELC Usage in Frame Transmission and Reception through SCI3_1

2. Description of Modules Used

2.1 Event Link Controller (ELC)

The features of the ELC are described below. Figure 4 shows a block diagram of the ELC.

The event link controller (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.

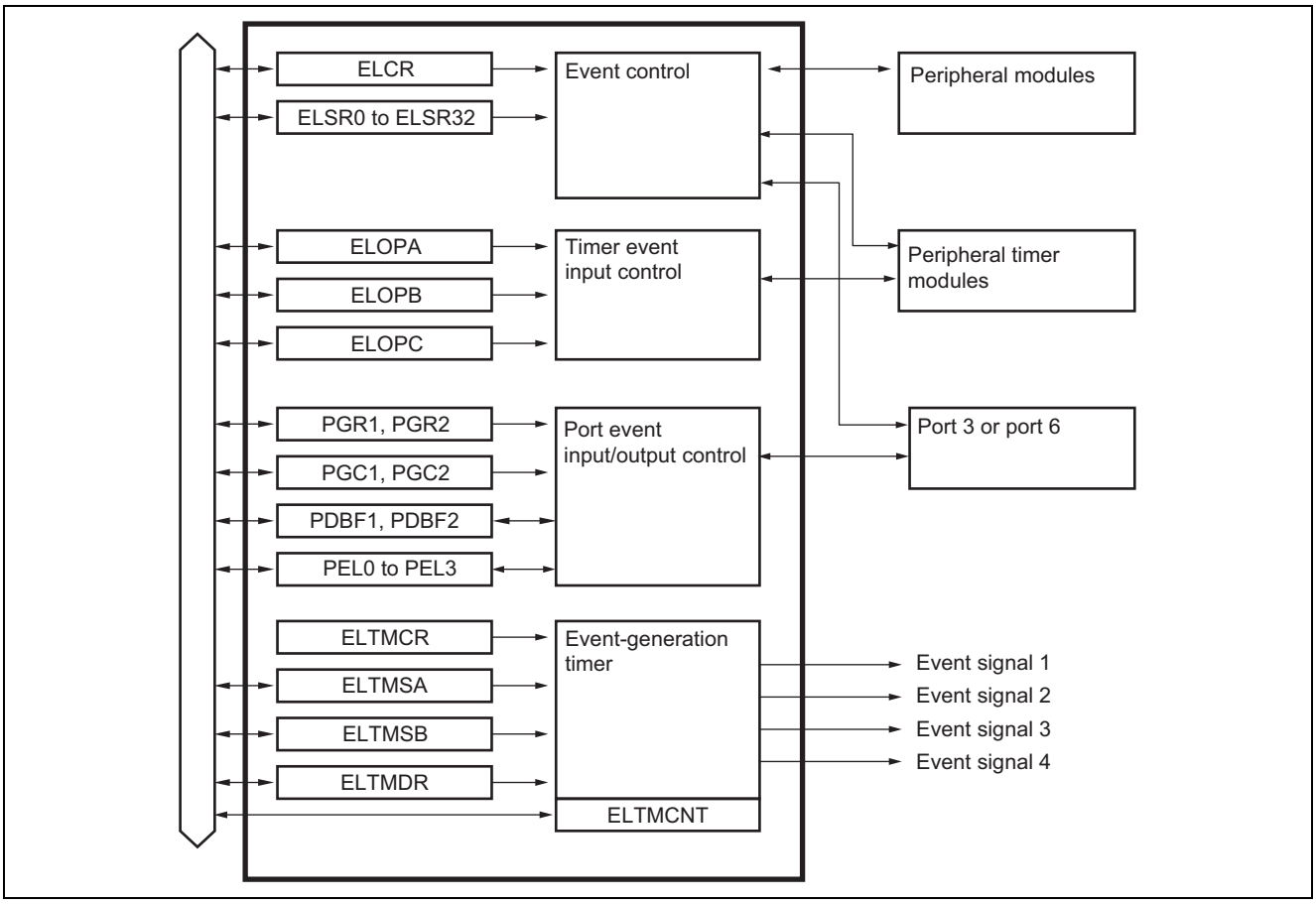


Figure 4 Block Diagram of Event Link Controller

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 clocked 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 5 and 6 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 clocked synchronous serial communications mode
- 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: 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	SMR	H'FF0550	Available
		RXD	BRR	H'FF0551	
		TXD	SCR3	H'FF0552	
			TDR	H'FF0553	
			SSR	H'FF0554	
			RDR	H'FF0555	
			RSR	—	
			TSR	—	
			SPMR	H'FF0556	
Channel 2	SCI3_2*[2]	SCK3_2	SMR_2	H'FF0558	Available
		RXD_2/IrRxD	BRR_2	H'FF0559	
		TXD_2/IrTxD	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	SMR_3	H'FF0560	Available
		RXD_3	BRR_3	H'FF0561	
		TXD_3	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.

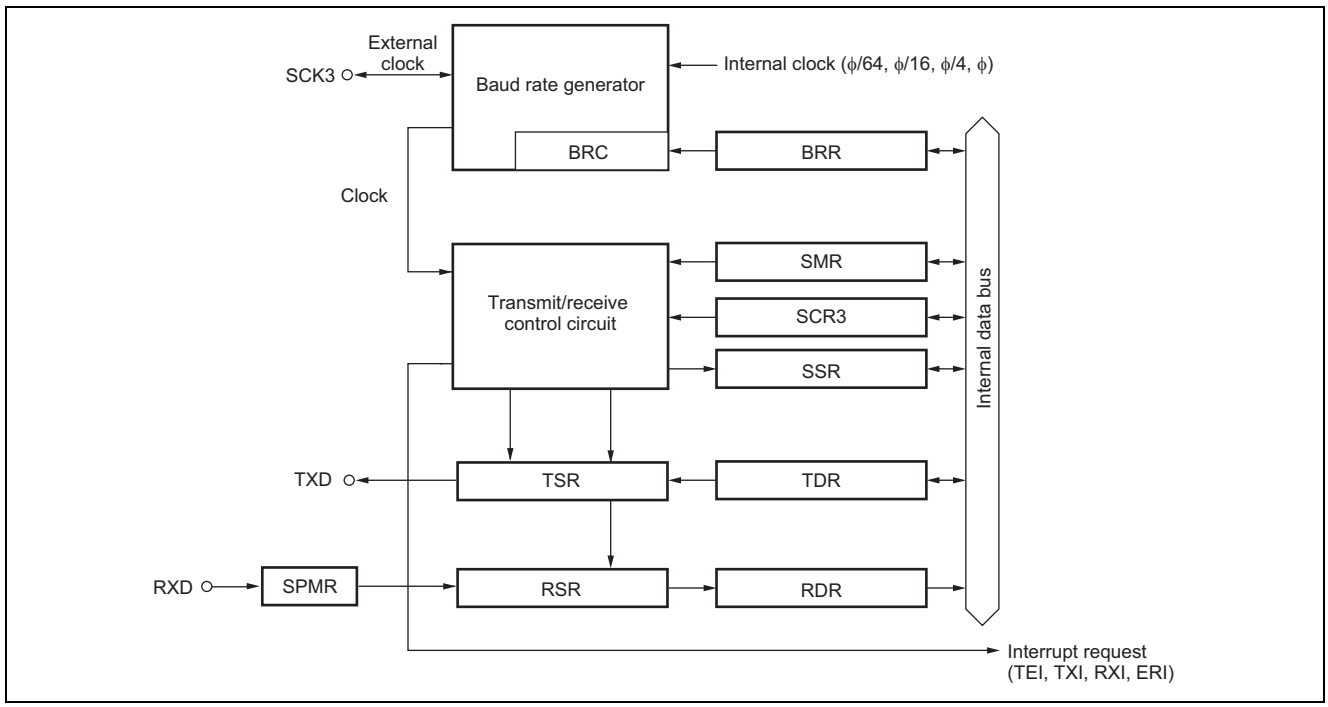


Figure 5 Block Diagram of SCI3 and SCI3_3

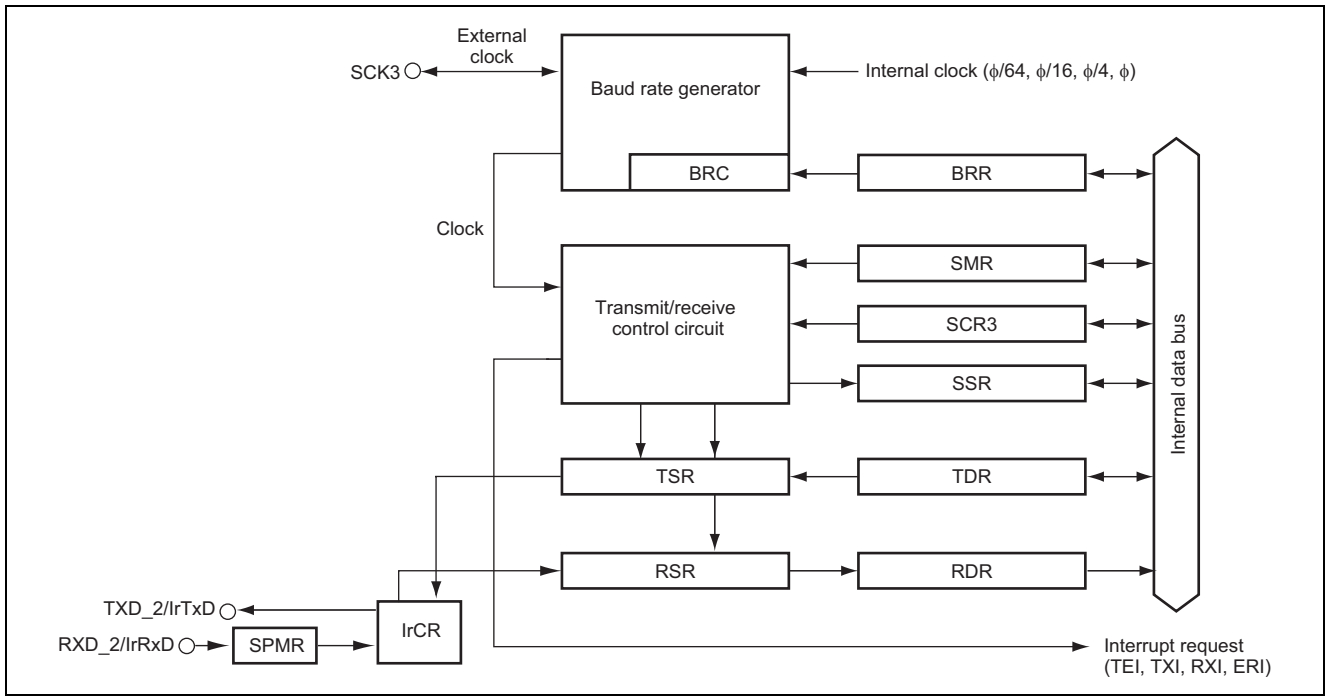


Figure 6 Block Diagram of SCI3_2

2.3 Data Transfer Controller (DTC)

The features of the DTC are described below.

This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software to transfer data.

Figure 7 shows a block diagram of the DTC.

- Transfer possible over any number of channels
- Three transfer modes
  (1) Normal mode
  One operation transfers one byte or one word of data.
  Memory address is incremented or decremented by 1 or 2.
  From 1 to 65,536 transfers can be specified.
  (2) Repeat mode
  One operation transfers one byte or one word of data.
  Memory address is incremented or decremented by 1 or 2.
  Once the specified number of transfers (1 to 256) has ended, the initial state is restored, and transfer is repeated.
  (3) Block transfer mode
  One operation transfers specified one block of data.
  From 1 to 65,536 transfers can be specified.
  Either the transfer source or the transfer destination is designated as a block area.
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 16-Mbyte address space is possible.
- Activation by software is possible.
- Transfer can be set in byte or word units.
- A CPU interrupt can be requested for the interrupt that activated the DTC.
- Module standby mode can be set.

The DTC's register information is stored in the on-chip RAM. A 32-bit bus connects the DTC to the on-chip RAM, enabling 32-bit/1-state reading and writing of the DTC register information.

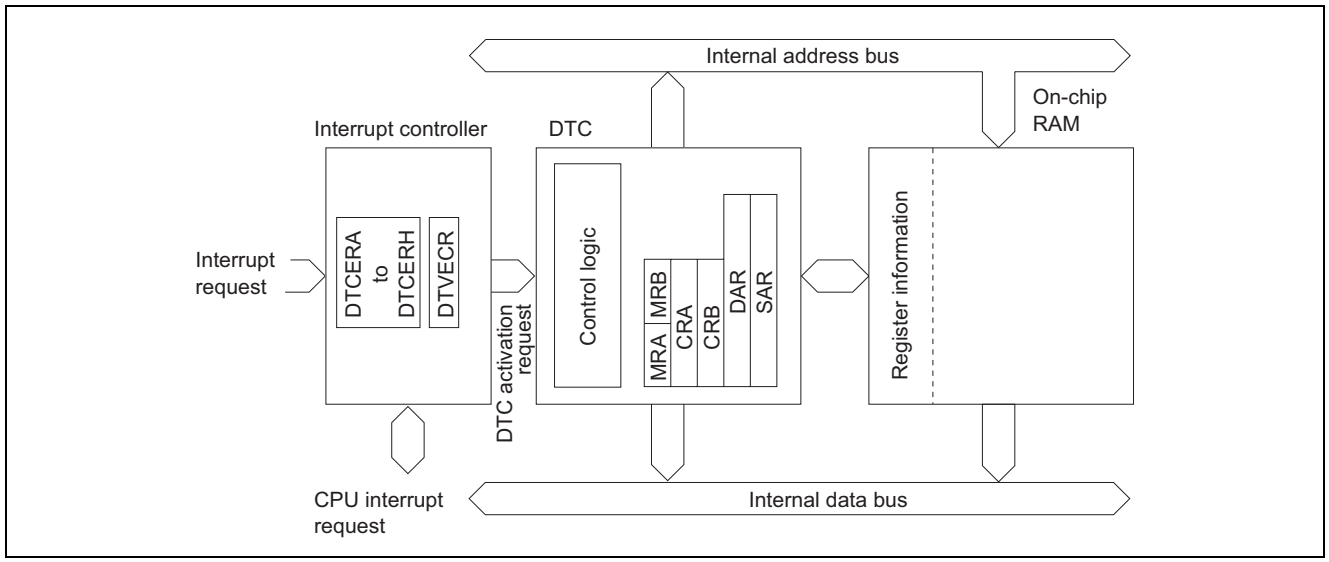


Figure 7 Block Diagram of DTC

2.3.1 Activation Sources

The DTC operates when activated by an interrupt request or by writing to DTVECR by software. An interrupt request can be designated by the DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source flag is the RDRF flag of SCI3_1.

When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities for the interrupt sources. Table 2 shows a relationship between activation sources and DTCER clearing conditions. Figure 8 shows a block diagram of DTC activation source control. For details, see the section on the interrupt controller of the *H8S/20103, H8S/20203, H8S/20223 Group Hardware Manual* (REJ09B0465).

Table 2 Relationship between Activation Sources and DTCER Clearing

Activation Source	DISEL = 0 and Specified Number of Transfers Has Not Ended	DISEL = 1 or Specified Number of Transfers Has Ended
Activation by software	The SWDTE bit is cleared to 0.	• The SWDTE bit retains the value 1. • Interrupt request to the CPU
Activation by an interrupt	• The corresponding bit of DTCER retains the value 1. • Activation source flag is cleared to 0.	• The corresponding bit of the DTCER bit is cleared to 0. • Activation source flag retains the value 1. • The interrupt that had been the source for activation is issued as an interrupt request for the CPU.

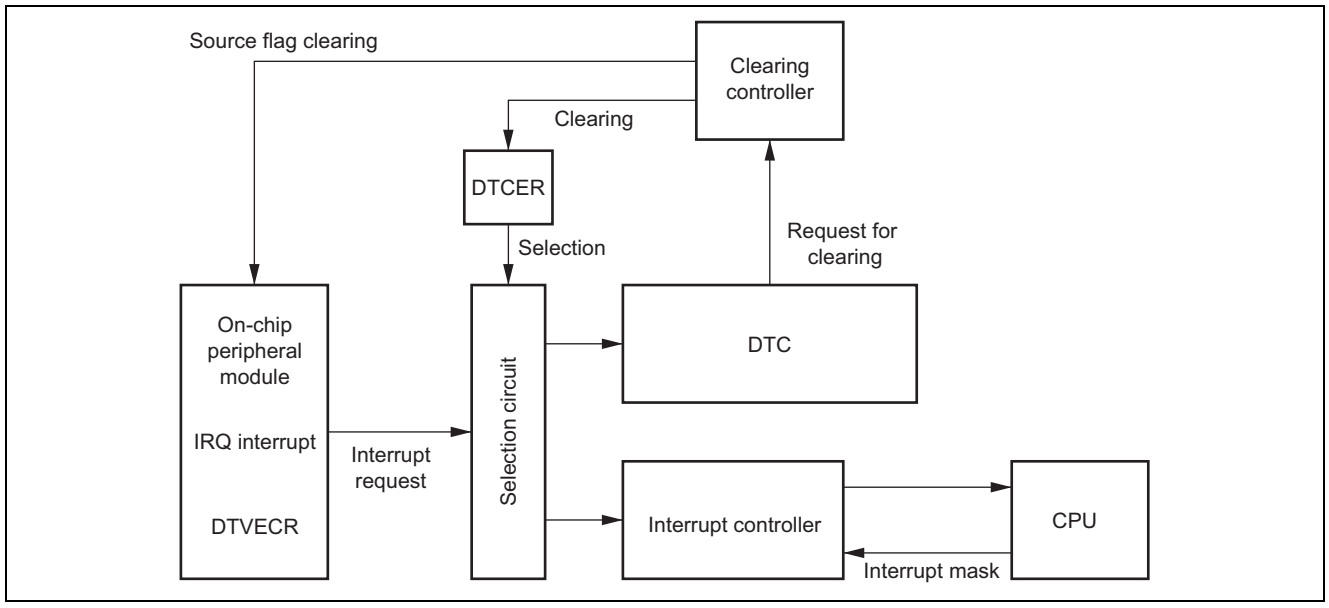


Figure 8 Block Diagram of DTC Activation Source Control

2.3.2 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM. Register information should be located at the address that is multiple of four. Locating the register information in address space is shown in figure 9. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information. The register information start address should be located at the corresponding vector address to the activation source. Figure 10 shows correspondences between the DTC vector address and register information. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

Table 3 gives a list of interrupt sources capable of DTC activation, addresses in the vector table, and the corresponding DTCE bits.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if VOFR and DTVECR are H'0000 and H'18 respectively, the vector address is H'0430.

The configuration of the vector address is a 2-byte unit. These two bytes specify the lower bits of the start address. Variable vector addresses can be used by setting VOFR. For details on VOFR settings, see the section on the interrupt controller of the *H8S/20103, H8S/20203, H8S/20223 Group Hardware Manual* (REJ09B0465).

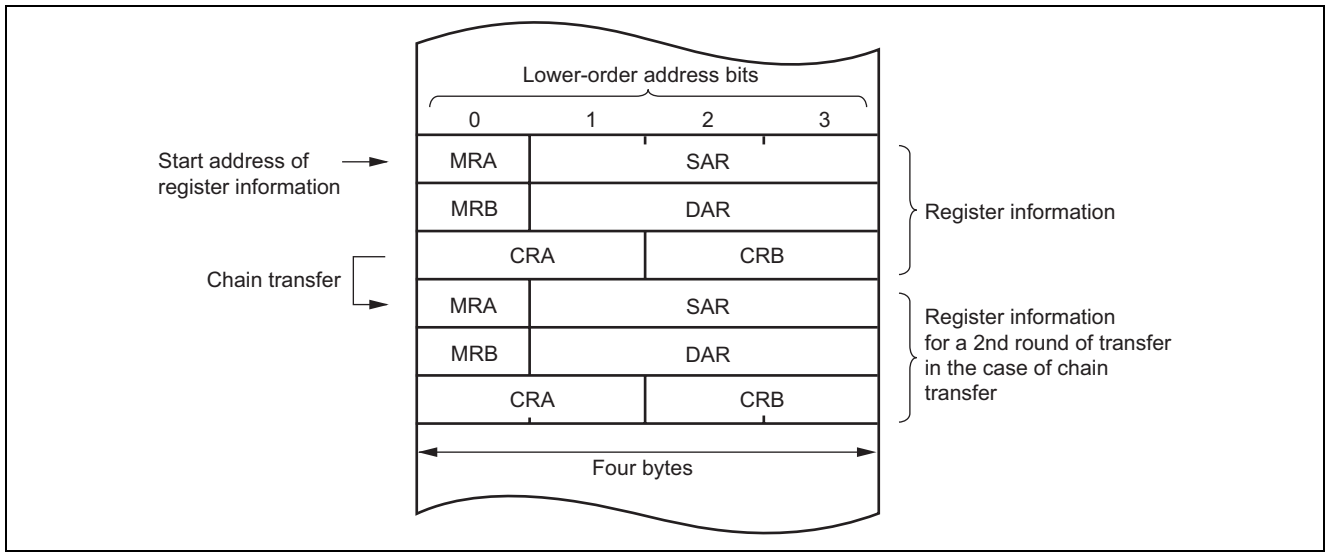


Figure 9 Locating DTC Register Information in Address Space

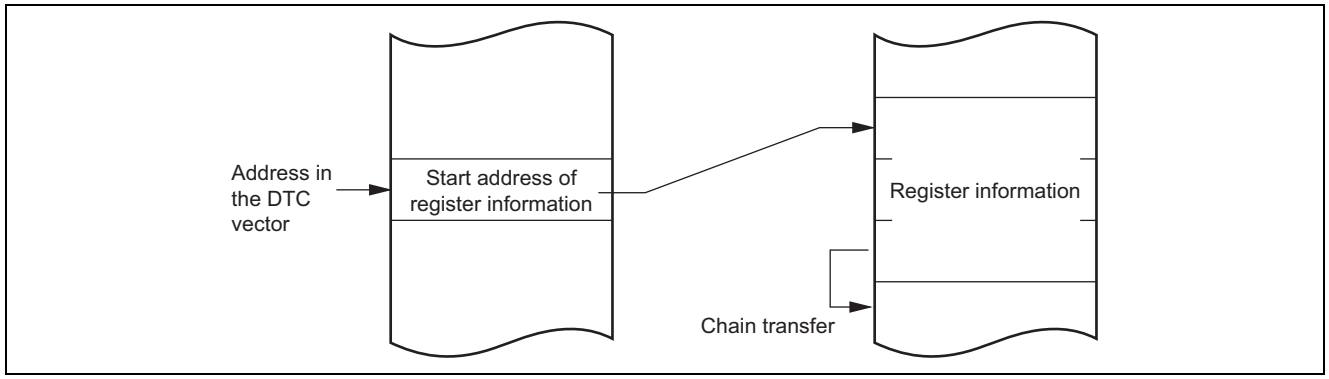


Figure 10 Correspondence between the Address in a DTC Vector and Register Information

Table 3 Interrupt Sources, Addresses of DTC Vectors, and Corresponding DTCE Bits

Origin of Activation Source	Activation Source	Vector Number	Address in Vector Table*[1]	DTCE*[5]	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (DTVECR[6:0] × 2)	—	High
External pin	IRQ0	22	H'42C to H'42D	DTCEA7	
	IRQ1	23	H'42E to H'42F	DTCEA6	
	IRQ2	24	H'430 to H'431	DTCEA5	
	IRQ3	25	H'432 to H'433	DTCEA4	
	IRQ4	26	H'434 to H'435	DTCEA3	
	IRQ5	27	H'436 to H'437	DTCEA2	
	IRQ6	28	H'438 to H'439	DTCEA1	
	IRQ7	29	H'43A to H'43B	DTCEA0	
A/D converter unit 1	IADEND_1 (conversion completion)	30	H'43C to H'43D	DTCEB7	
	IADCMP_1 (compare condition match)	31	H'43E to H'43F	DTCEB6	
A/D converter unit 2	IADEND_2 (conversion completion)	32	H'442 to H'443	DTCEB5	
	IADCMP_2 (compare condition match)	33	H'444 to H'445	DTCEB4	
ELC	ELC1FP (ELSR12 event occurrence)	35	H'446 to H'447	DTCEB3	
	ELC2FP (ELSR30 event occurrence)	36	H'448 to H'449	DTCEB2	
SCI3 channel 1	SCI3_1 RXI	38	H'44C to H'44D	DTCEB1	
	SCI3_1 TXI	39	H'44E to H'44F	DTCEB0	
SCI3 channel 2	SCI3_2 RXI	42	H'454 to H'455	DTCEC7	
	SCI3_2 TXI	43	H'456 to H'457	DTCEC6	
SCI3 channel 3	SCI3_3 RXI	46	H'45C to H'45D	DTCEC5	
	SCI3_3 TXI	47	H'45E to H'45F	DTCEC4	
IIC2/SSU	IIC2/SSU_RXI	60	H'478 to H'479	DTCED7	
	IIC3/SSU_TXI	61	H'47A to H'47B	DTCED6	
Timer RC*[3]	ITCMA Input capture A/compare match A	71	H'48E to H'48F	DTCED3	
	ITCMB Input capture B/compare match B	72	H'490 to H'491	DTCED2	
	ITCMC Input capture C/compare match C	73	H'492 to H'493	DTCED1	
	ITCMD Input capture D/compare match D	74	H'494 to H'495	DTCED0	
Timer RD unit 0 channel 0	ITDMA0_0 Input capture A/compare match A	76	H'498 to H'499	DTCEE7	
	ITDMB0_0 Input capture B/compare match B	77	H'49A to H'49B	DTCEE6	
	ITDMC0_0 Input capture C/compare match C	78	H'49C to H'49D	DTCEE5	
	ITDMD0_0 Input capture D/compare match D	79	H'49E to H'49F	DTCEE4	Low

Origin of Activation Source	Activation Source	Vector Number	Address in Vector Table*[1]	DTCE*[5]	Priority
Timer RD unit 0 channel 1*[4]	ITDMA0_1 Input capture A/compare match A	82	H'4A4 to H'4A5	DTCEE3	High
	ITDMB0_1 Input capture B/compare match B	83	H'4A6 to H'4A7	DTCEE2	
	ITDMC0_1 Input capture C/compare match C	84	H'4A8 to H'4A9	DTCEE1	
	ITDMD0_1 Input capture D/compare match D	85	H'4AA to H'4AB	DTCEE0	
Timer RD unit 1 channel 2*[4]	ITDMA1_2 Input capture A/compare match A	87	H'4AE to H'4AF	DTCEF7	
	ITDMB1_2 Input capture B/compare match B	88	H'4B0 to H'4B1	DTCEF6	
	ITDMC1_2 Input capture C/compare match C	89	H'4B2 to H'4B3	DTCEF5	
	ITDMD1_2 Input capture D/compare match D	90	H'4B4 to H'4B5	DTCEF4	
Timer RD unit 1 channel 3*[4]	ITDMA1_3 Input capture A/compare match A	93	H'4BA to H'4BB	DTCEF3	
	ITDMB1_3 Input capture B/compare match B	94	H'4BC to H'4BD	DTCEF2	
	ITDMC1_3 Input capture C/compare match C	95	H'4BE to H'4BF	DTCEF1	
	ITDMD1_3 Input capture D/compare match D	96	H'4C0 to H'4C1	DTCEF0	
Timer RE	ITESC	100	H'4C8 to H'4C9	DTCEG4	
	ITEMI	101	H'4CA to H'4CB	DTCEG3	
	ITEHR	102	H'4CC to H'4CD	DTCEG2	
	ITEDY	103	H'4CE to H'4CF	DTCEG1	
	ITEWK	104	H'4D0 to H'4D1	DTCEG0	
Timer RG	ITGMA Input capture A/compare match A	109	H'4DA to H'4DB	DTCEH3	
	ITGMB Input capture B/compare match B	110	H'4DC to H'4DD	DTCEH2	Low

Notes: 1. “Address in vector table” indicates the 11 lower-order bits of the address in the vector table when VOFR = H'0000.
2. Supported only in the H8S/20223 Group and reserved in other products.
3. Supported only in the H8S/20103 Group and reserved in other products.
4. Not supported in the H8S/20103 Group and reserved in other products.
5. The DTCE bits with no corresponding interrupt are reserved. The write value should always be 0.

3. Principle of Operation

Figure 11 shows how the ELC is employed in frame transmission by SCI3. This proceeds by means of the hardware and software processing described in figure 11. The transmit-data-empty signal is generated each time the TDRE bit becomes set to 1. The signal activates the DTC without CPU intervention. The DTC proceeds to write the next byte for transmission to TDR. Figure 12 shows how DTC transfer proceeds in the transmission phase of this sample task.

Figure 13 shows how the ELC is employed in frame reception by SCI3. This proceeds by means of the hardware and software processing described in figure 13. The receive-data-full signal is generated each time the RDRF bit becomes set to 1. The signal activates the DTC without CPU intervention. The DTC proceeds to store the received byte in a received data buffer which has been allocated in RAM. Figure 14 shows how DTC transfer proceeds in the reception phase of this sample task .

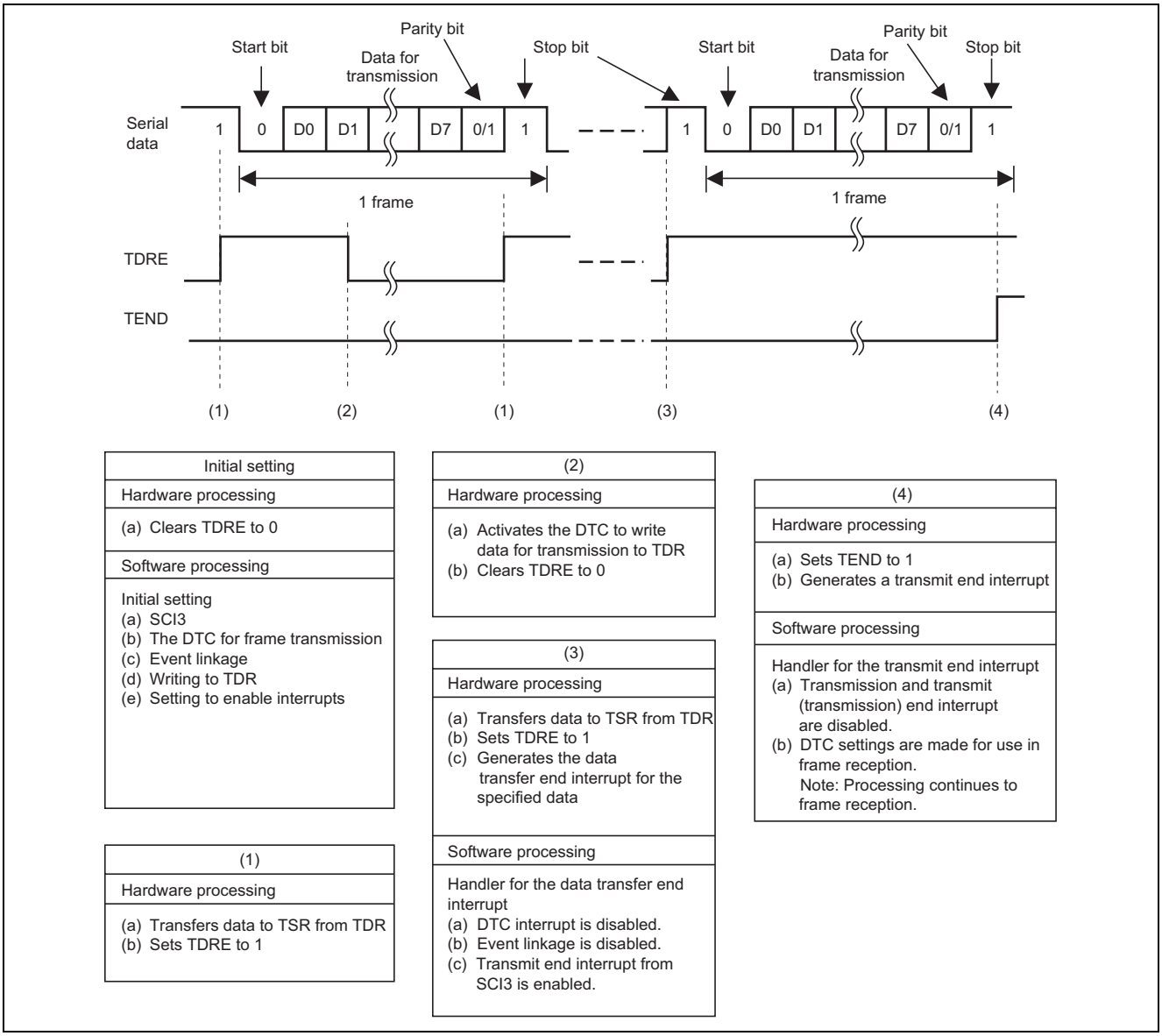


Figure 11 Principle of ELC Usage in Frame Transmission by SCI3 in This Sample Task

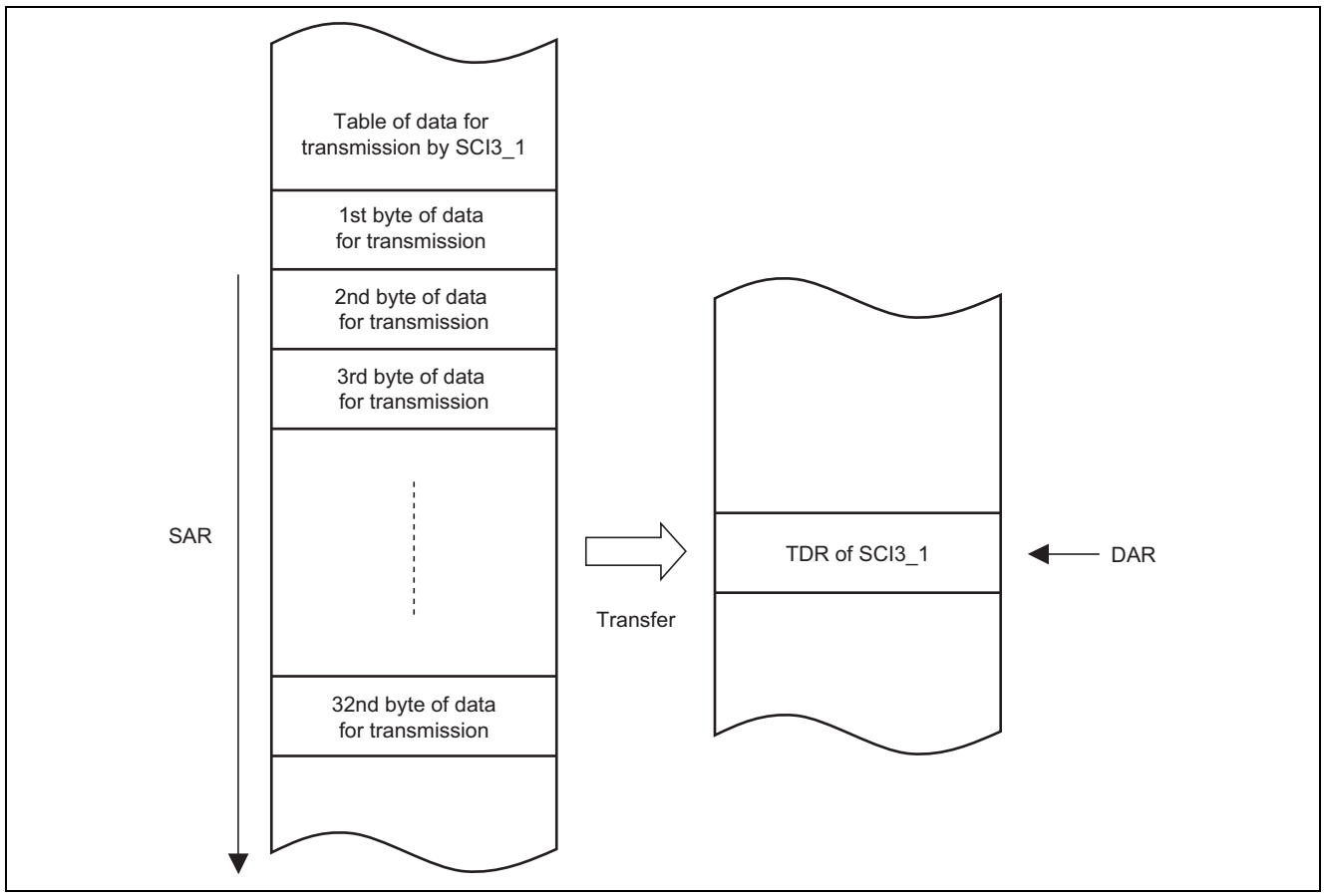
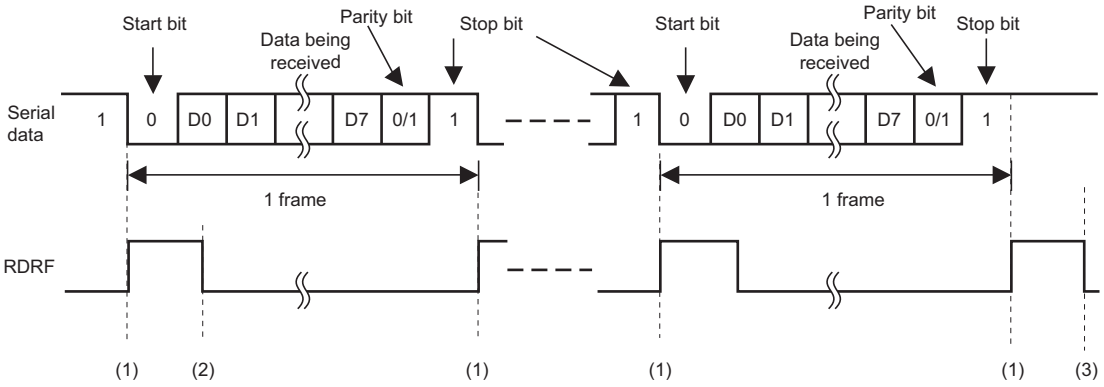


Figure 12 DTC Operation in Case of SCI3 Frame Transmission by Using the ELC in This Sample Task



(1)
Hardware processing
(a) Transfers data to RDR from RSR (b) Sets RDRF to 1

(2)
Hardware processing
(a) Activates the DTC to transfer received data from RDR to the received-data buffer (b) Clears RDRF to 0

(3)
Hardware processing
(a) Activates the DTC to store received data to receive-data buffer from RDR (b) Clears RDRF to 0 (c) Generates the data transfer end interrupt for the specified data
Software processing
Handler for the data transfer end interrupt (a) DTC interrupt is disabled. (b) Event linkage is disabled. (c) Reception is disabled.

Figure 13 Principle of ELC Usage in Frame Reception by SCI3 in This Sample Task

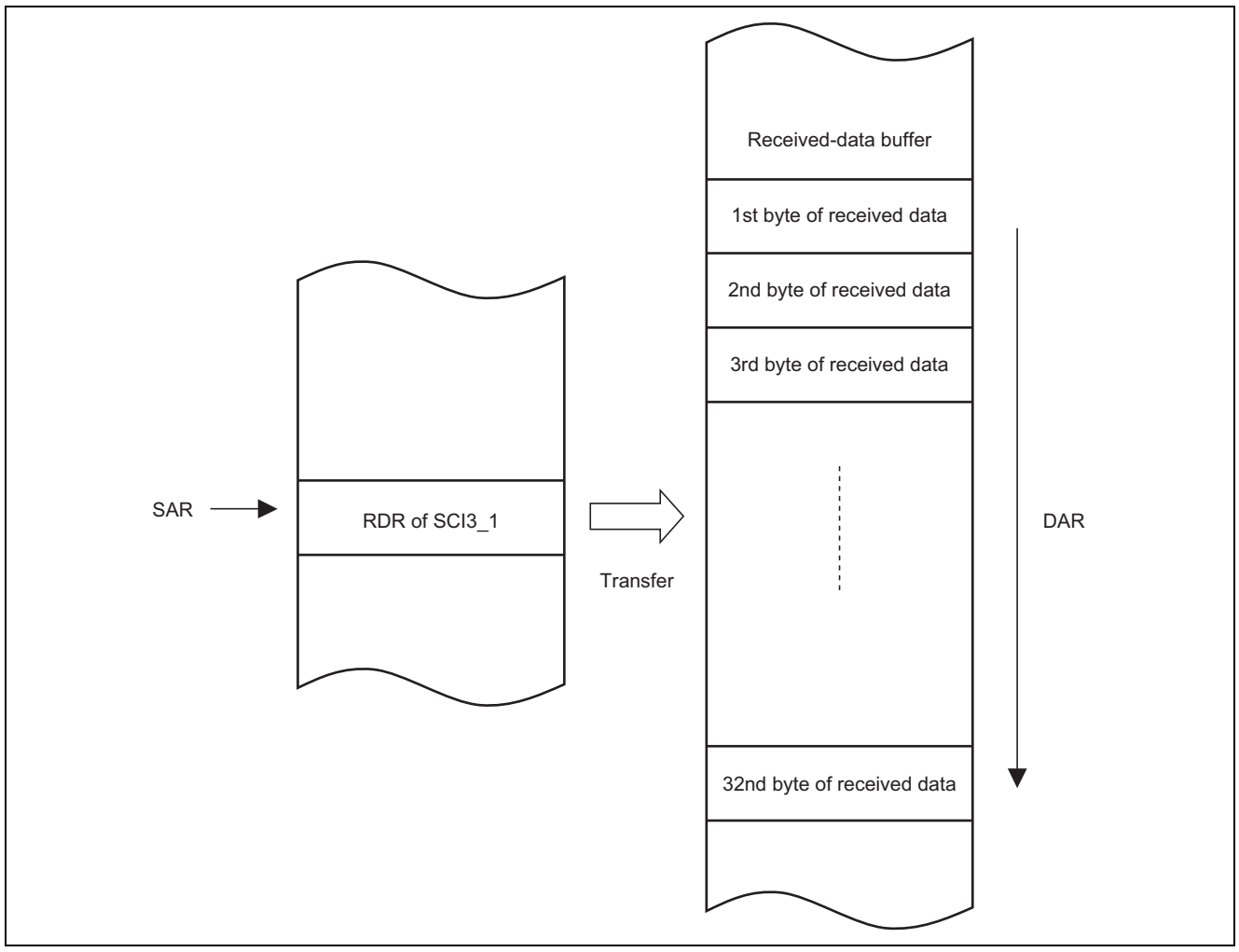


Figure 14 DTC Operation in Case of SCI3 Frame Reception by Using the ELC in This Sample Task

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 various other functions and enables interrupts.
System initialization routine	h8s_sysinit	Makes settings for module standby, system clock and bus-master operating clock, and halts the WDT.
SCI3_1 setting routine	init_sci31	Makes settings for SCI3 channel 1.
DTC setting routine for frame transmission	init_trs_dtc	Makes DTC and ELC settings for frame transmission.
DTC setting routine for frame reception	init_rcv_dtc	Makes settings for DTC and ELC for frame reception.
SCI3_1 transmission start routine	trs_st_sci31	Starts transmission through SCI3 channel 1 transmission.
ELSR30 event interrupt routine	INT_ELC2FP_ELC	Clears interrupt request flags, and disables ELSR30 event interrupts and event linkage. Enables the transmission end interrupt in the case of frame transmission. Disables reception in the case of frame reception.
SCI3_1 transmission end interrupt routine	INT_TEI_SCI31	Disables transmission and transmission end interrupts, and calls the init_rcv_dtc function.

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	PMR22	The P22 pin is set to the TXD pin.	H'FF0001	1
	PMR21	The P21 pin is set to the RXD pin.		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	Pull-up MOS of the P21 pin is set to ON.	H'FF0011	1
ELCSR	ELIE2	The ELF2 interrupt is enabled.	H'FF0528	1
	ELF2	The ELSR30 event flag is cleared.		0
DTCERB	ELC2FP	The ELSR30 event interrupt source is selected as the source for DTC activation.	H'FF0535	1
DTVECR	DTVEC6 to DTVEC0	DTC activation vector numbers are specified.	H'FF053D	B'0000000
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	Selects even parity.		0
	STOP	1 stop bit is selected.		0
	MP	The multiprocessor communications function is disabled.		0
	CKS[1:0]	Bit rate is set to 9,600 bps with BRR.		B'00
BRR		Bit rate is set to 9,600 bps with the CKS[1:0] bits in SMR.	H'FF0551	64
SCR3	TIE	The TXI interrupt request is disabled.	H'FF0552	0
	RIE	RXI and ERI interrupt requests are disabled.		0
	TE	Transmission is enabled.		1
	RE	Reception is enabled.		1
	TEIE	The TEI interrupt request is enabled.		1
	CKE[1:0]	Specifies on-chip baud rate generator as the clock source.		B'00
TDR		Data for transmission are written.	H'FF0553	Arbitrary

Register Name	Symbol	Description	Address	Setting
SSR	TDRE	[Setting conditions] • The TE bit in SCR3 being 0 • Data having been transferred from TDR to TSR [Clearing conditions] • The CPU writing 0 to the bit after reading it as 1 • The CPU writing data for transmission to TDR • The DTC transferring data to TDR in response to a TXI interrupt request while the DTC settings satisfying the flag clearing conditions*[1]	H'FF0554	0 or 1
	RDRF	[Setting condition] • Reception ending normally and received data being transferred from RSR to RDR [Clearing conditions] • The CPU writing 0 to RDRF after reading it as 1 • The CPU reading data from RDR • The DTC transferring data from RDR with an RXI interrupt request and the DTC settings satisfying the flag clearing conditions *[1]		0 or 1
	TEND	[Setting conditions] • The TE bit in SCR3 being 0 • The TDRE bit being 1 on transmission of the last bit of a character for transmission [Clearing conditions] • 0 being written to the TDRE bit after it has been read as 1 • Writing of data for transmission to TDR		0 or 1
RDR		Received data are read.	H'FF0555	Undifined
ELSR30		Settings are made to link DTC activation with the compare match A signal from channel 0 of timer RD1.	H'FF069E	H'4A or H'4C
ELCR	ELCON	Linkage of all the events are enabled.	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 clock source ϕbase.		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 period of timer ϕosc oscillation settling time.	H'FF06D5	H'0E
TMWD		Clock input to WDT is prohibited.	H'FFFF99	H'F7
TCSRWD		Writing to TMWD is controlled.	H'FFFF9A	H'A3
MSTCR1	MSTWDT	Watchdog timer is released from module standby.	H'FFFFDC	0
	MSTDTC	The DTC is released from module standby.		0
MSTCR2	MSTSCI3_1	SCI3 channel 1 is released from module standby.	H'FFFFDD	0

Register Name	Symbol	Function	Address	Setting
MRA*[2]	SM[1:0]	In frame transmission, the SAR is incremented after transfer. In frame reception, the SAR is unchanged after transfer.	H'FFDF80	B'10 B'00
	DM[1:0]	In frame transmission, the DAR is unchanged after transfer. In frame reception, the DAR is incremented after transfer.		B'00 B'10
	MD[1:0]	The DTC is set to normal mode.		B'00
	Sz	Byte-size transfer		0
SAR*[2]		Transfer source address is specified. In frame transmission, address in the table of data for transmission by SCI3_1 In frame reception, address of RDR of SCI3_1	H'FFDF81	H'000A63 H'FF0555
MRB*[2]	CHNE	Setting is made so that transfer is not chained.	H'FFDF84	0
	DISEL	Setting is made so that an interrupt request for the CPU is only generated when the specified data transfer is completed.		0
DAR*[2]		Transfer destination address is specified. In frame transmission, address in TDR of SCI3_1 In frame reception, address of receive-data buffer of SCI3_1	H'FFDF85	H'000553 H'FFDF8C
CRA*[2]		Number of unit transfers is specified. In frame transmission, frame size – 1 In frame reception, frame size	H'FFDF88	31 32
CRB*[2]		—	H'FFDF8A	—

Notes: 1. The DTC clears the peripheral module flags when all of the following three conditions are satisfied:
   a. The DISEL bit is 0.
   b. The value in the transfer counter (count register CRA in normal and repeat modes or count register CRB in block mode) is not 0.
   c. A chain transfer is not used.
2. Information for the DTC registers is located in RAM.

4.4 RAM Usage

Table 6 gives a description of RAM usage in this sample task (information for the DTC registers is also located in RAM).

Table 6 RAM Usage

Label Name	Description	Memory	Name of Employing Module
rcv_buf[32]	Receive-data buffer of SCI3_1	1 byte × 32	—

4.5 Description of Definition in Use

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

Table 7 Description of Definition in Use

Label Name	Description	Constant
FRAME_SIZE	Setting of number of frames for both transmission and reception	32

4.6 Description of Constants

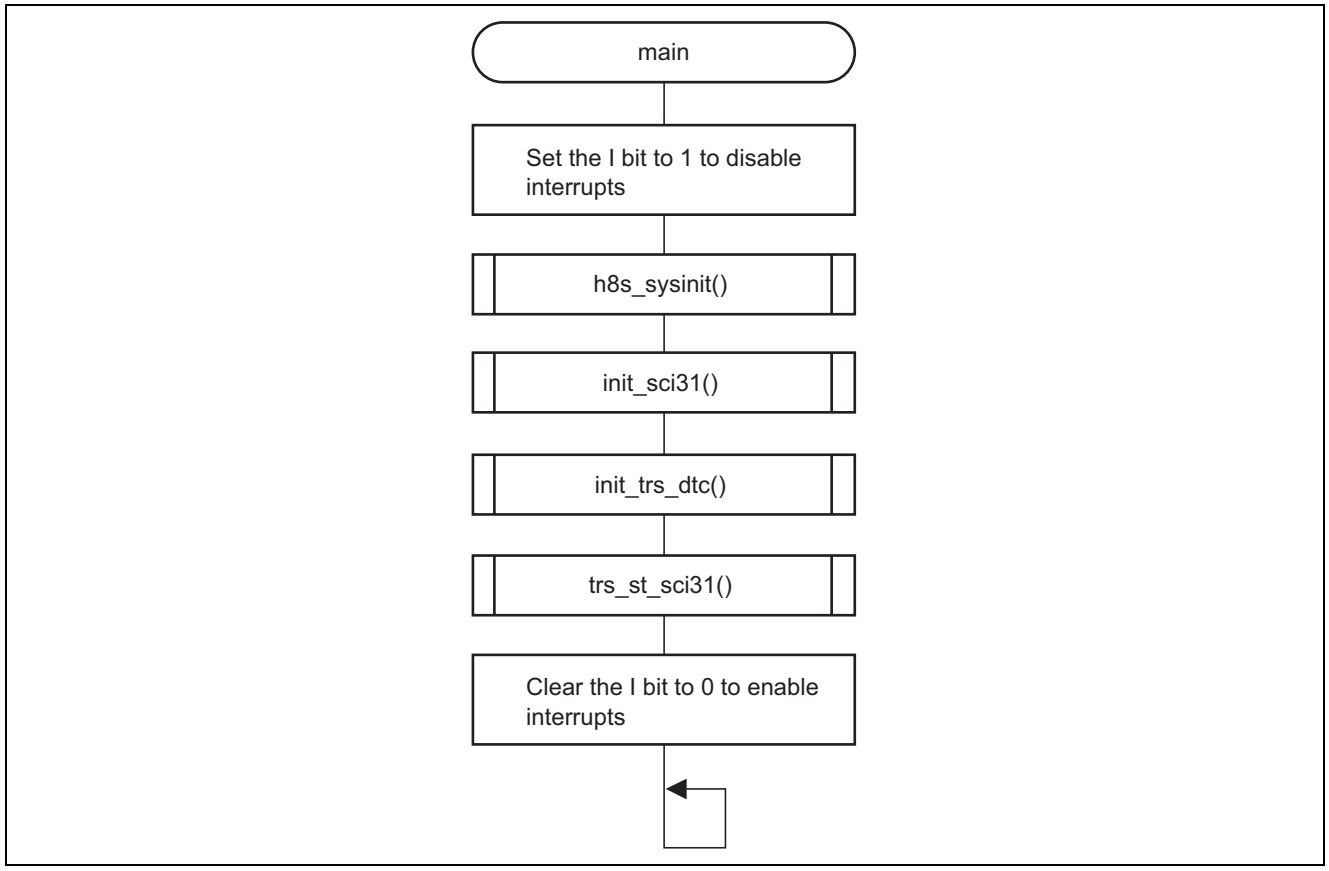
Table 8 gives description of the constants used in this sample task.

Table 8 Description of Constants

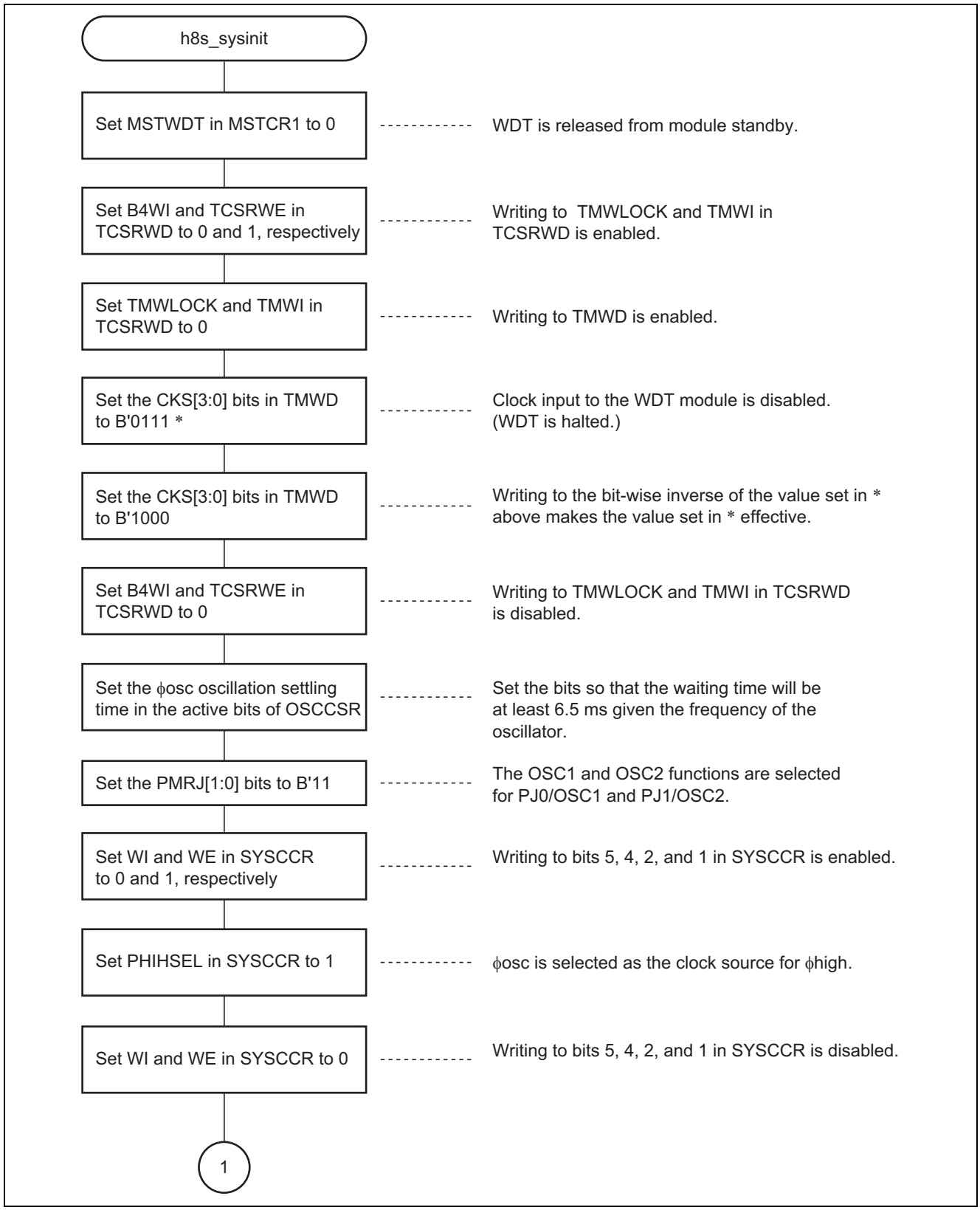
Label Name	Description	Address	Constant
dtcvect_tbl[]	Start address of information of the DTC registers is selected for the address in the DTC vector table corresponding to the DTC activation source.	H'000448	H'DF80
trs_buf[0] to trs_buf[31]	Transmit-data table of SCI3_1	H'000A62	H'11, H'22, H'33, H'44, H'55, H'66, H'77, H'88, H'99, H'AA, H'BB, H'CC, H'DD, H'EE, H'FF, H'FE, H'DC, H'BA, H'98, H'76, H'54, H'32, H'10, H'FC, H'DB, H'CA, H'B9, H'A8, H'97, H'86, H'75, H'64

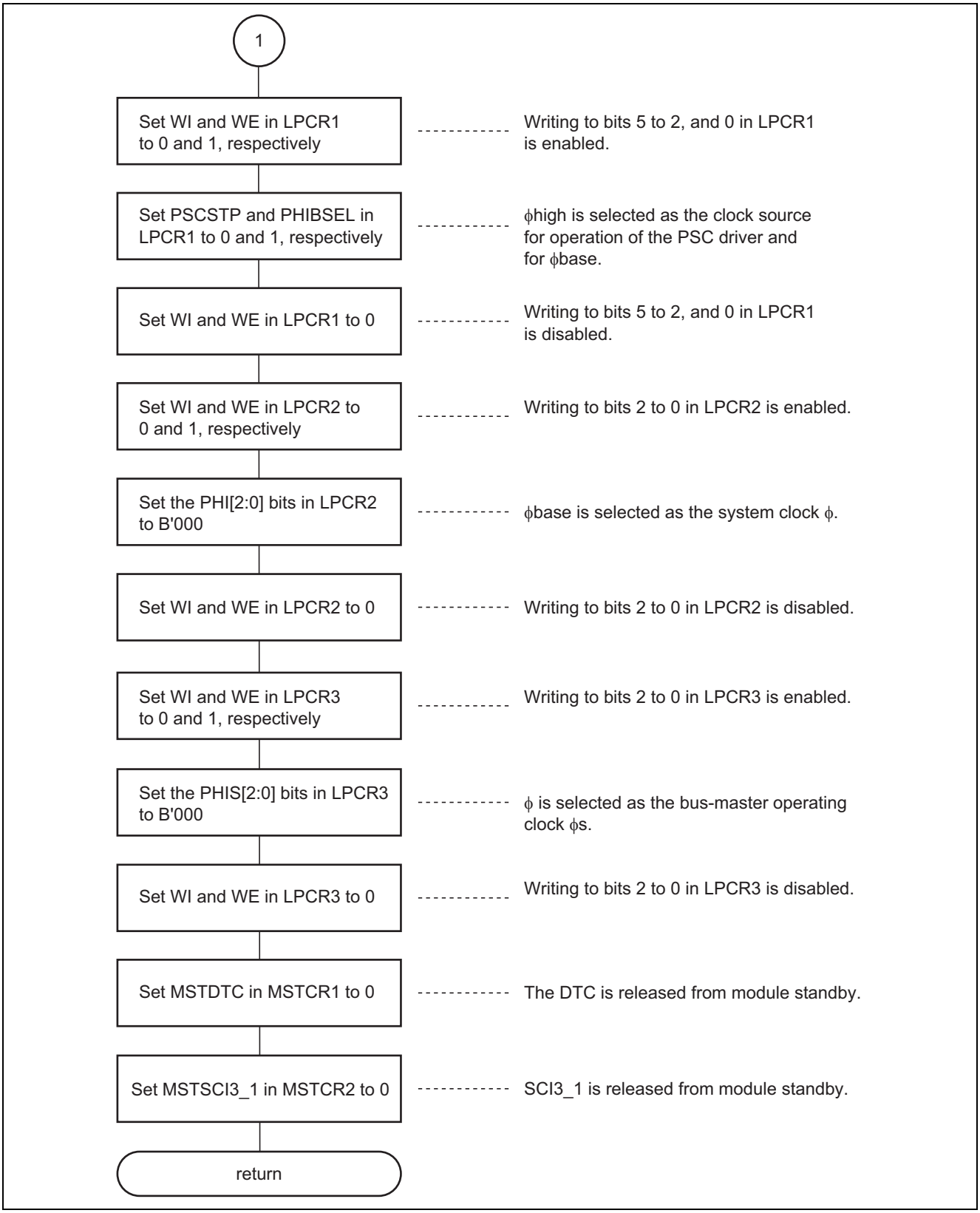
5. Flowcharts

5.1 Main Routine

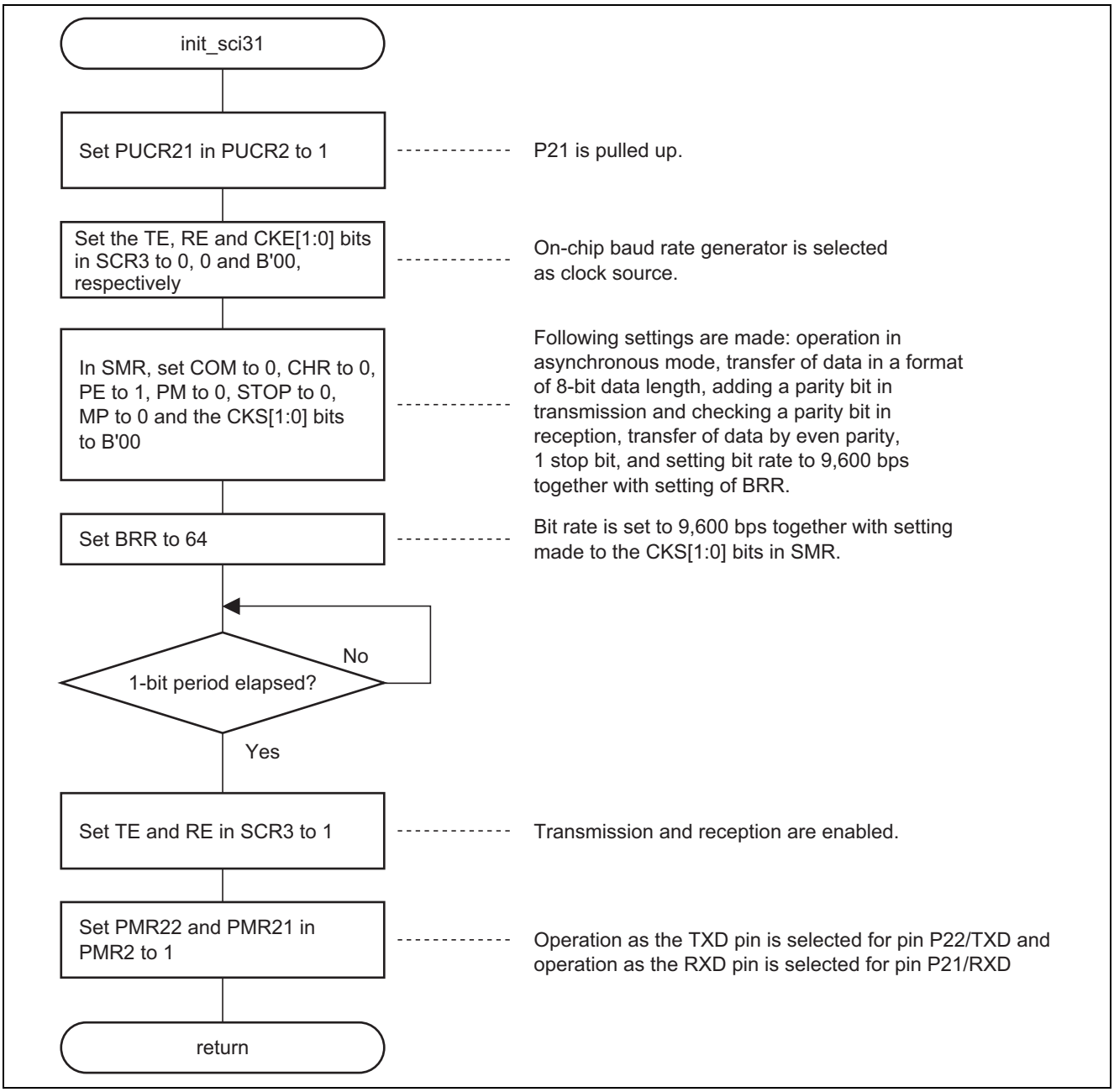


5.2 System Initialization Routine

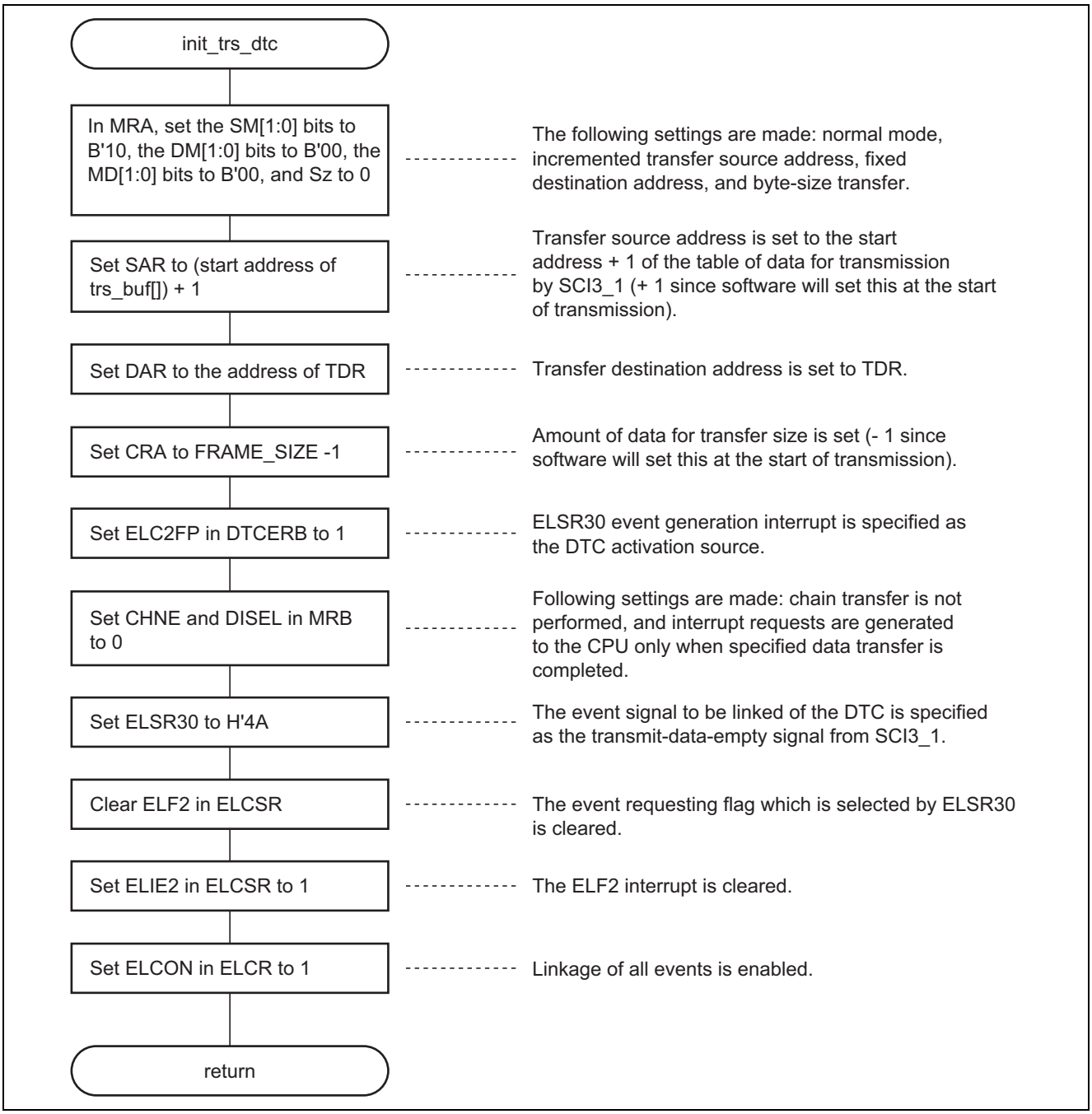



1
Set WI and WE in LPCR1 to 0 and 1, respectively
Writing to bits 5 to 2, and 0 in LPCR1 is enabled.
Set PSCSTP and PHIBSEL in LPCR1 to 0 and 1, respectively
ϕhigh is selected as the clock source for operation of the PSC driver and for ϕbase.
Set WI and WE in LPCR1 to 0
Writing to bits 5 to 2, and 0 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 MSTDTC in MSTCR1 to 0
The DTC is released from module standby.
Set MSTSCI3_1 in MSTCR2 to 0
SCI3_1 is released from module standby.
return

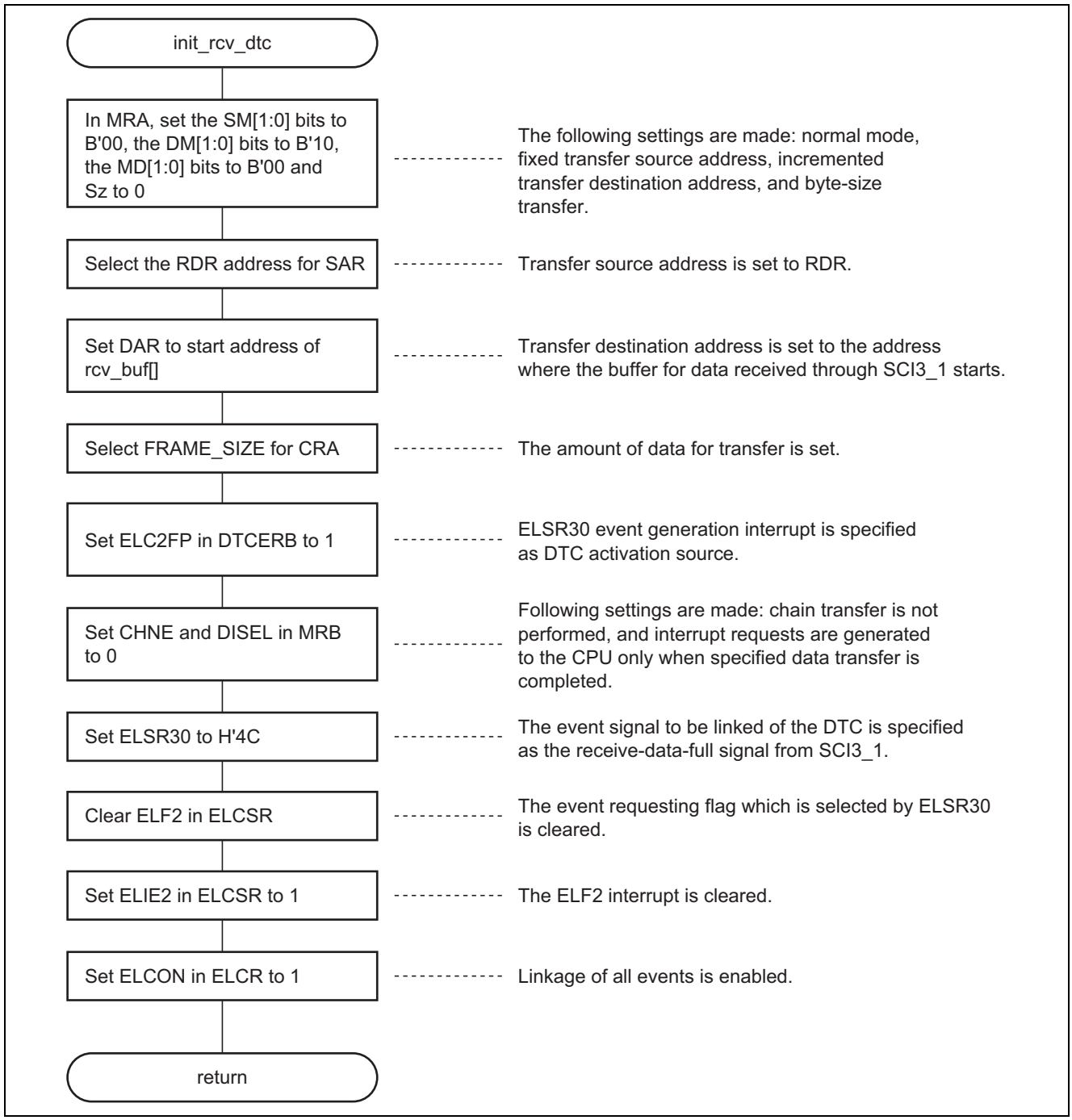
5.3 SCI3_1 Setting Routine



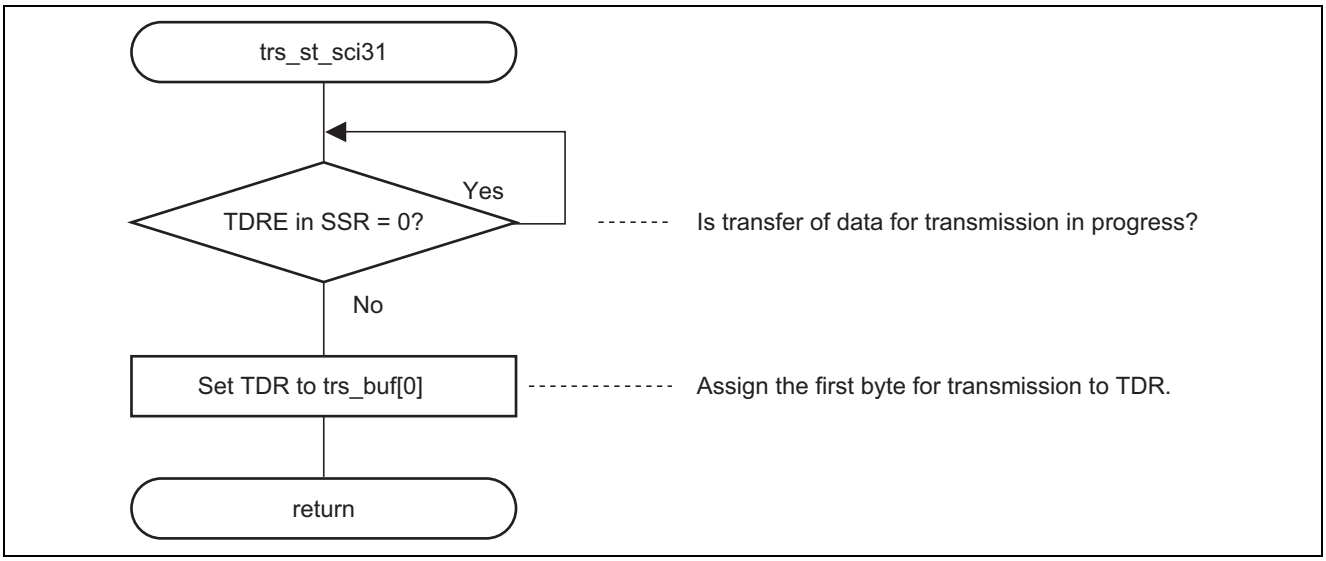
5.4 DTC Setting Routine for Frame Transmission



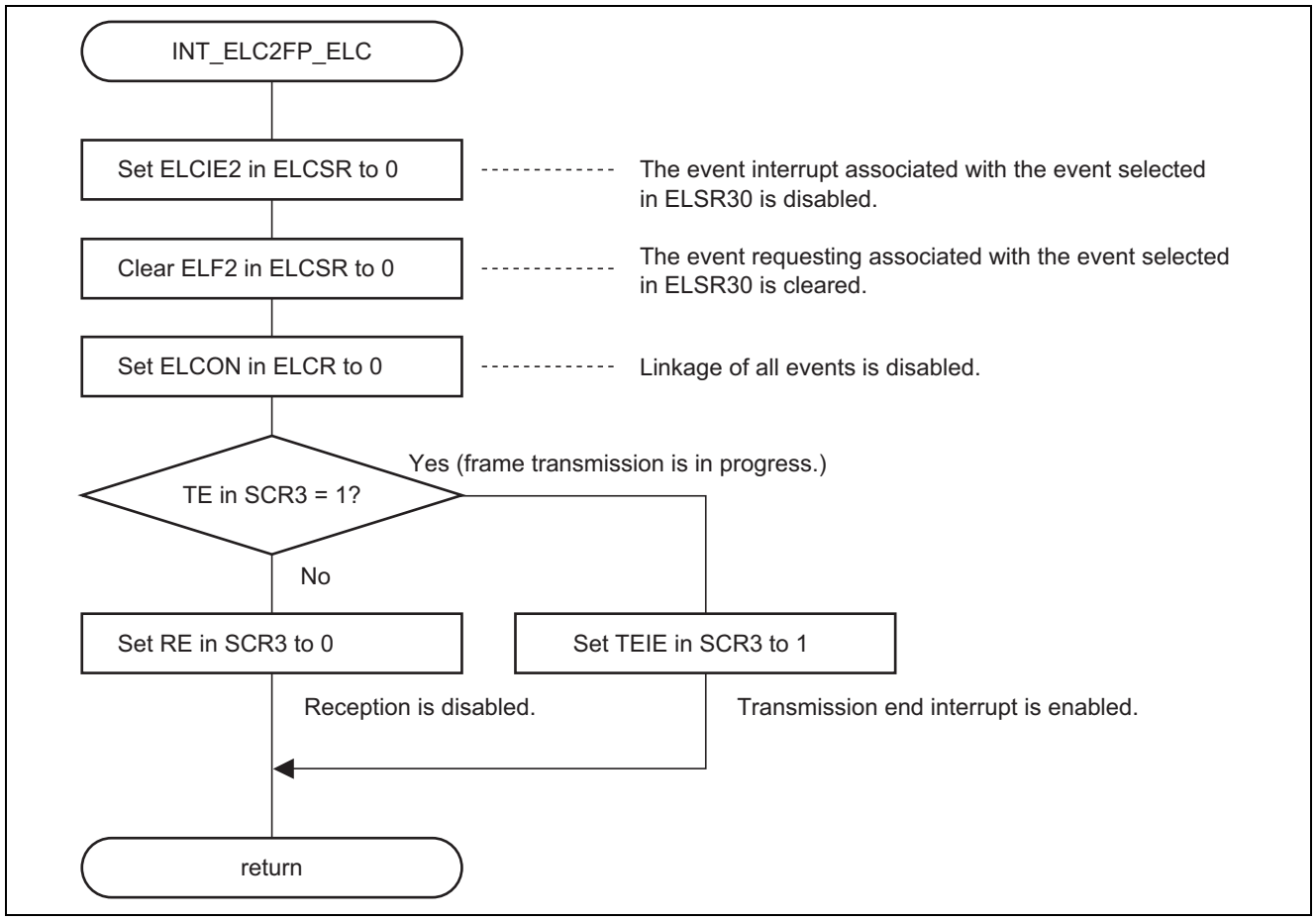
5.5 DTC Setting Routine for Frame Reception



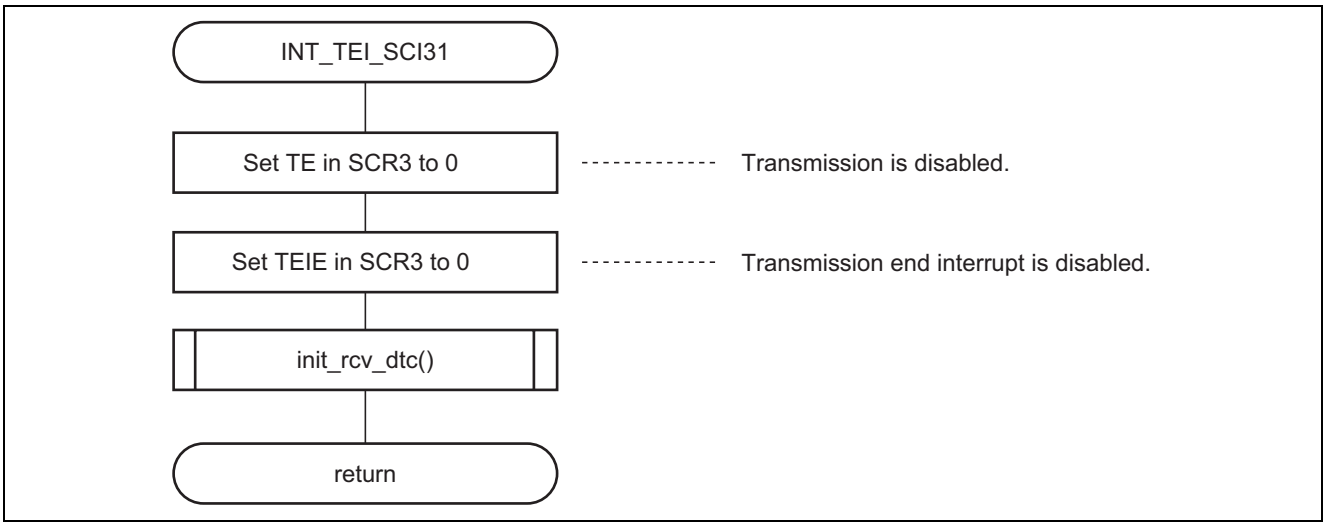
5.6 SCI3_1 Transmission Start Routine



5.7 ELSR30 Event Generation Interrupt Routine



5.8 SCI3_1 Transmission End Interrupt Routine



6. Program Listing

```
<elc_sci3_frame.c>
/************************************************************************/
/* H8S/2000 Tiny Series -H8S/20203-                                     */
/* Application Note                                                     */
/*                                                                      */
/* SCI3 frame sending and receiving by ELC                              */
/*                                                                      */
/* Function                                                             */
/* : start DTC by SCI3_1 transmit data empty, receive data full         */
/* Event Link DTC and SCI3_1                                            */
/*                                                                      */
/* External Clock : 20MHz                                               */
/* Internal Clock : 20MHz                                               */
/************************************************************************/
#include <machine.h>
#include "iodefine.h"

typedef struct
{
    union{
            unsigned char      MRA;                    /* DTC mode register A */
            struct{
                    unsigned long      dummy1:8;       /* dummy1 data (MRA address) */
                    unsigned long      SAR:24;         /* DTC source address register */
            }SAR;
    }UN_MRA_SAR;
    union{
            unsigned char      MRB;                    /* DTC mode register B */
            struct{
                    unsigned long      dummy2:8;       /* dummy2 data (DAR address) */
                    unsigned long      DAR:24;         /* DTC destination address register */
            }DAR;
    }UN_MRB_DAR;
    union{
            unsigned short     CRA;                    /* DTC transfer count register A */
            struct{
                    unsigned char      CRAH;           /* DTC transfer count register BH */
                    unsigned char      CRAL;           /* DTC transfer count register BL */
            }CHAR;
    }UN_CRA;
    union{
            unsigned short     CRB;                    /* DTC transfer count register B */
            struct{
                    unsigned char      CRBH;           /* DTC transfer count register BH */
                    unsigned char      CRBL;           /* DTC transfer count register BL */
            }CHAR;
    }UN_CRB;
} st_dtc_reg;
```

```
/******************************************************/
/* Definition of const data                           */
/******************************************************/
/* SCI3_1 */
#define FRAME_SIZE        32         /* frame size */

/* transmit data */
const unsigned char trs_buf[FRAME_SIZE]={
           0x11, 0x22, 0x33, 0x44,
           0x55, 0x66, 0x77, 0x88,
           0x99, 0xAA, 0xBB, 0xCC,
           0xDD, 0xEE, 0xFF, 0xFE,
           0xDC, 0xBA, 0x98, 0x76,
           0x54, 0x32, 0x10, 0xFC,
           0xDB, 0xCA, 0xB9, 0xA8,
           0x97, 0x86, 0x75, 0x64
};

/******************************************************/
/* Declaration of function prototypes                 */
/******************************************************/
void main(void);
void init_sci31(void);
void init_trs_dtc(void);
void init_rcv_dtc(void);
void trs_st_sci31(void);
void h8s_sysinit(void);

/******************************************************/
/* Definition of RAM area                             */
/******************************************************/
#pragma section DTC
st_dtc_reg DTC_REG;                  /* DTC registers */

#pragma section
unsigned char rcv_buf[FRAME_SIZE];    /* reception buffer */

/******************************************************/
/* Name: main                                         */
/* Parameters: None                                   */
/* Returns: None                                      */
/* Description: User main                             */
/******************************************************/
void main(void)
{
    unsigned char ii;

    set_ccr(0x80);                   /* set CCR-Ibit */

    h8s_sysinit();                   /* initialize system */

    init_sci31();                    /* initialize SCI3_1 */
```

```
    init_trs_dtc();                 /* initialize DTC (SCI3_1 transmit) */

    trs_st_sci31();                 /* transmit start SCI3_1 */

    set_imask_ccr(0);               /* interrupt enable */

    while(1);
}

/*****************************************************/
/* Name: init_sci31                                  */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: initialize SCI3_1                    */
/*****************************************************/
void init_sci31(void)
{
    unsigned short wait;

    /* 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;                /* bitrate => 9600 bit/s */

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

    SCI3_1.SCR3.BYTE |= 0x30;       /* set TE, RE */
    PMR2.BYTE |= 0x06;              /* P21=>RXD, P22=>TXD */
}

/*****************************************************/
/* Name: init_trs_dtc                                */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: initialize DTC (SCI3_1 transmit)     */
/*****************************************************/
void init_trs_dtc(void)
{
    DTC_REG.UN_MRA_SAR.MRA = 0x80;          /* normal mode, SAR increment, DAR hold */
                                            /* transfer 1byte size */

    DTC_REG.UN_MRA_SAR.SAR.SAR = (unsigned long)&trs_buf[1]; /* Address of source for transfer*/
    DTC_REG.UN_MRB_DAR.DAR.DAR = (unsigned long)&SCI3_1.TDR; /* Address of destination for transfer*/
    DTC_REG.CRA = FRAME_SIZE - 1;           /* Set transfer counter */

    DTC.DTCERB.BIT.ELC2FP = 1;              /* DTC start by ELSR30 event */
    DTC_REG.UN_MRB_DAR.MRB = 0x00;          /* disable chain, interrupt transfer end */
```

```
    /* Set event link, SCI3_1 transmit data empty */
    ELC.ELSR30.BYTE = 0x4A;
    INTC.ELCSR.BYTE &= ~0x02;            /* clear ELF2 */
    INTC.ELCSR.BIT.ELCIE2 = 1;           /* ELF2 interrupt enable */
    ELC.ELCR.BIT.ELCON = 1;              /* event link enable */
}

/*****************************************************/
/* Name: init_rcv_dtc                                */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: initialize DTC (SCI3_1 receive)      */
/*****************************************************/
void init_rcv_dtc(void)
{
    DTC_REG.UN_MRA_SAR.MRA = 0x20;       /* normal mode, SAR hold, DAR increment */
                                         /* transfer 1byte size */

    DTC_REG.UN_MRA_SAR.SAR.SAR = (unsigned long)&SCI3_1.RDR; /* Forwarding former address */
    DTC_REG.UN_MRB_DAR.DAR.DAR = (unsigned long)rcv_buf;     /* Address at forwarding destination */
    DTC_REG.CRA = FRAME_SIZE;            /* Set transfer counter */

    DTC.DTCERB.BIT.ELC2FP = 1;           /* DTC start by ELSR30 event */
    DTC_REG.UN_MRB_DAR.MRB = 0x00;       /* disable chain, interrupt transfer end */

    /* Set event link, SCI3_1 receive data full */
    ELC.ELSR30.BYTE = 0x4C;
    INTC.ELCSR.BYTE &= ~0x02;            /* clear ELF2 */
    INTC.ELCSR.BIT.ELCIE2 = 1;           /* ELF2 interrupt enable */
    ELC.ELCR.BIT.ELCON = 1;              /* event link enable */
}

/*****************************************************/
/* Name: trs_st_sci31                                */
/* Parameters: None                                  */
/* Returns: None                                     */
/* Description: start transmit SCI3_1                */
/*****************************************************/
void trs_st_sci31(void)
{
    while ( SCI3_1.SSR.BIT.TDRE == 0 );  /* wait when transmit not empty */

    SCI3_1.TDR = trs_buf[0];
}
```

```
/*****************************************************/
/* 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.MSTDTC = 0;                              /* DTC module standby off */
    MSTCR2.BIT.MSTSCI3_1 = 0;                           /* SCI3_1 module standby off */
}
```

```
<intprg.c>
/*****************************************************/
/* Extern Declaration of function prototype          */
/*****************************************************/
extern void init_rcv_dtc(void);

// vector 36 ELSR30 event ELC
__interrupt(vect=36) void INT_ELC2FP_ELC(void) {
    INTC.ELCSR.BIT.ELCIE2 = 0;            /* ELF2 interrupt disable */
    DTC.DTCERB.BIT.ELC2FP = 0;            /* clear ELC2FP */
    ELC.ELCR.BIT.ELCON = 0;               /* event link disable */

    /* transmit ? */
    if ( SCI3_1.SCR3.BIT.TE != 0 ){
            SCI3_1.SCR3.BIT.TEIE = 1;     /* transmit end interrupt enable */
    }
    /* receive */
    else{
            SCI3_1.SCR3.BIT.RE = 0;       /* receive disable */
    }
}

// vector 40 TEI SCI31
__interrupt(vect=40) void INT_TEI_SCI31(void) {
    SCI3_1.SCR3.BIT.TE = 0;               /* trasmit disable */
    SCI3_1.SCR3.BIT.TEIE = 0;             /* transmit end interrupt disable */

    init_rcv_dtc();                       /* initialize DTC (SCI3_1 receive) */
}
```

6.1 Designation of Linkage Addresses

Section Name	Address
CDTC_VECT	H'000400
PResetPRG, PIntPRG	H'000500
P, C, C$DSEC, C$BSEC,D	H'000800
BDTC, 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

All trademarks and registered trademarks are the property of their respective owners.

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.

© 2008. Renesas Technology Corp., All rights reserved.