

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.

H8/300H Series On-Chip Supporting Modules

Application Note

Renesas 16-bit Single-chip
Microcomputer

H8 Family / H8/300H Series

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

Introduction

The H8/3003, 3042 and 3002 are high-performance microcomputers of the 32-bit H8/300H series, and feature a host of built-in peripheral functions.

The devices incorporate a CPU, RAM, 16-bit integrated timer unit (ITU), programmable timing pattern controller (TPC), serial communication interface (SCI), DMA controller (DMAC) and other peripheral functions which make them suitable for a broad range of applications, in small to large systems.

This application note is arranged in two parts: a basic edition covering operation examples that employ individual peripheral functions, and an application edition which deals with examples of use with combinations of these functions.

Although the task programs, operation of circuits etc., dealt with in this application note have been checked, we advise you to conduct your own check of operations before using the products.

Contents

Section 1	H8/300H Series Application Note: Guide for Use	1
1.1	Organization of Basic Edition	1
1.2	Organization of Application Edition	3
Section 2	Basic Edition	5
2.1	Pulse Output	5
2.1.1	Specifications	5
2.1.2	Description of Functions Used	6
2.1.3	Description of Operations	7
2.1.4	Description of Software	8
2.1.5	Flowchart	9
2.1.6	Program List	10
2.2	Two-Phase Encoder Count	12
2.2.1	Specifications	12
2.2.2	Description of Functions Used	12
2.2.3	Description of Operation	12
2.2.4	Description of Software	12
2.2.5	Flowcharts	12
2.2.6	Program List	12
2.3	Pulse High and Low Width Measurement	12
2.3.1	Specifications	12
2.3.2	Description of Functions Used	12
2.3.3	Description of Operations	12
2.3.4	Description of Software	12
2.3.5	Flowcharts	12
2.3.6	Program List	12
2.4	Complementary PWM Three-Phase Output	12
2.4.1	Specifications	12
2.4.2	Description of Functions Used	12
2.4.3	Description of Operations	12
2.4.4	Description of Software	12
2.4.5	Flowcharts	12
2.4.6	Program List	12
2.5	Long Cycle Pulse Output	12
2.5.1	Specifications	12
2.5.2	Description of Functions Used	12
2.5.3	Description of Operations	12
2.5.4	Description of Software	12

2.5.5	Flowchart	12
2.5.6	Program List	12
2.6	Positive/Negative PWM Three-Phase Output	12
2.6.1	Specifications	12
2.6.2	Description of Functions Used	12
2.6.3	Description of Operations	12
2.6.4	Description of Software	12
2.6.5	Flowchart	12
2.6.6	Program List	12
2.7	PWM Five-Phase Output	12
2.7.1	Specifications	12
2.7.2	Description of Functions Used	12
2.7.3	Description of Operations	12
2.7.4	Description of Software	12
2.7.5	Flowchart	12
2.7.6	Program List	12
2.8	Timer Counter Simultaneous Reset with External Signal	12
2.8.1	Specifications	12
2.8.2	Description of Functions Used	12
2.8.3	Description of Operations	12
2.8.4	Description of Software	12
2.8.5	Flowchart	12
2.8.6	Program List	12
2.9	One-Shot Pulse Output Using Buffer Mode	12
2.9.1	Specifications	12
2.9.2	Description of Functions Used	12
2.9.3	Description of Operations	12
2.9.4	Description of Software	12
2.9.5	Flowcharts	12
2.9.6	Program List	12
2.10	4-bit × 4 System Output	12
2.10.1	Specifications	12
2.10.2	Description of Functions Used	12
2.10.3	Description of Operations	12
2.10.4	Description of Software	12
2.10.5	Flowcharts	12
2.10.6	Program List	12
2.11	Asynchronous SCI	12
2.11.1	Specifications	12
2.11.2	Description of Functions Used	12
2.11.3	Description of Operations	12
2.11.4	Description of Software	12

2.11.5	Flowcharts.....	12
2.11.6	Program List	12
2.12	Simultaneous Transmit/Receive Operation.....	12
2.12.1	Specifications.....	12
2.12.2	Description of Functions Used.....	12
2.12.3	Description of Operations.....	12
2.12.4	Description of Software	12
2.12.5	Flowcharts.....	12
2.12.6	Program List	12
2.13	Multiprocessor Communication.....	12
2.13.1	Specifications.....	12
2.13.2	Description of Functions Used.....	12
2.13.3	Description of Operations.....	12
2.13.4	Description of Software	12
2.13.5	Flowcharts.....	12
2.13.6	Program List	12
2.14	Block Transfer	12
2.14.1	Specifications.....	12
2.14.2	Description of Functions Used.....	12
2.14.3	Description of Operations.....	12
2.14.4	Description of Software	12
2.14.5	Flowcharts.....	12
2.14.6	Program List	12
Section 3 Application Edition.....		12
3.1	High-Speed Data Output.....	12
3.1.1	Specifications.....	12
3.1.2	Method.....	12
3.1.3	Description of Functions Used.....	12
3.1.4	Description of Operations.....	12
3.1.5	Description of Software	12
3.1.6	Flowcharts.....	12
3.1.7	Program List	12
3.2	SCI Continuous Transmit/Receive.....	12
3.2.1	Specifications.....	12
3.2.2	Method.....	12
3.2.3	Description of Functions Used.....	12
3.2.4	Description of Software	12
3.2.5	Flowcharts.....	12
3.2.6	Program List	12
3.3	Sine Wave Output.....	12
3.3.1	Specifications.....	12

3.3.2	Method.....	12
3.3.3	Description of Functions Used.....	12
3.3.4	Description of Operations.....	12
3.3.5	Description of Software.....	12
3.3.6	Flowcharts.....	12
3.3.7	Program list.....	12
3.4	Four-Phase Stepping Motor Application Example.....	12
3.4.1	Specifications.....	12
3.4.2	Method.....	12
3.4.3	Description of Functions Used.....	12
3.4.4	Description of Operations.....	12
3.4.5	Description of Software.....	12
3.4.6	Flowcharts.....	12
3.4.7	Program List.....	12
3.5	Five-Phase Stepping Motor Application Example.....	12
3.5.1	Specifications.....	12
3.5.2	Method.....	12
3.5.3	Description of Functions Used.....	12
3.5.4	Description of Operations.....	12
3.5.5	Description of Software.....	12
3.5.6	Flowcharts.....	12
3.5.7	Program List.....	12

Section 1 H8/300H Series Application Note: Guide for Use

This application note is arranged in two parts as shown in figure 1.1.

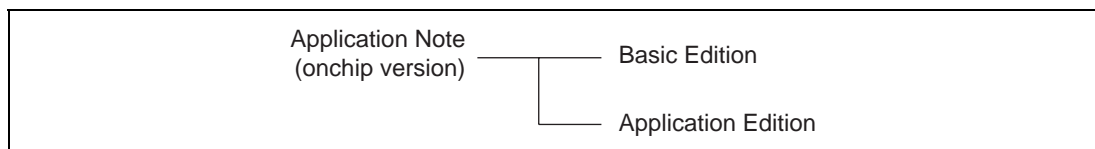


Figure 1.1 Application Note Organization

Basic Edition: Describes H8/300H peripheral function operations based on examples of single tasks.

Application Edition: Describes H8/300H operations based on tasks implemented by a combination of peripheral functions.

1.1 Organization of Basic Edition

The basic edition is arranged as in figure 1.2, and describes methods of use for single peripheral functions.

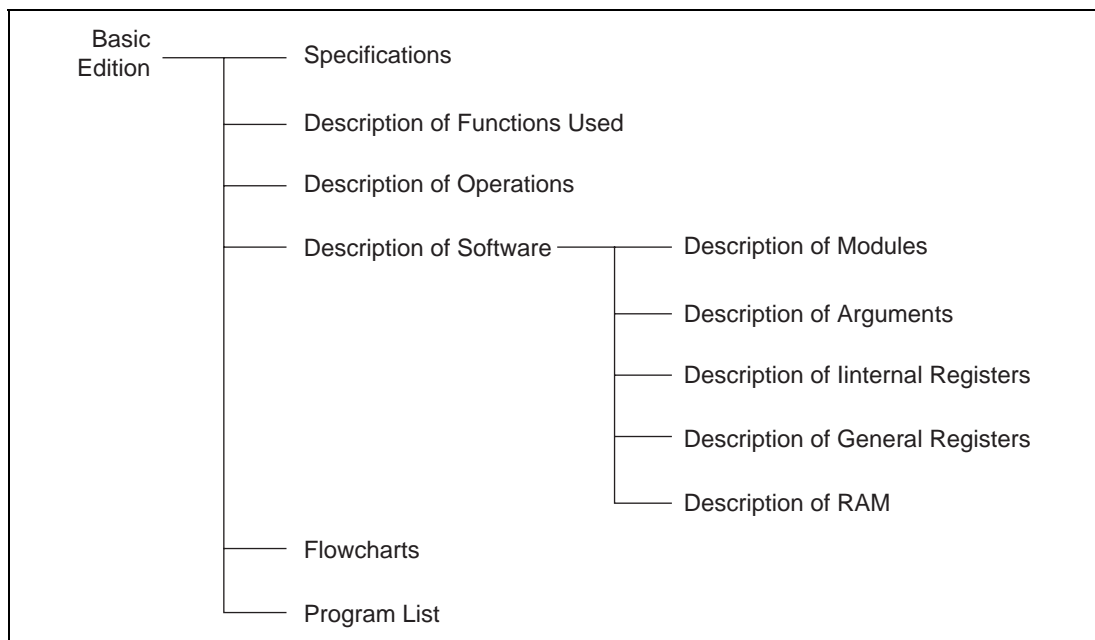


Figure 1.2 Organization of Basic Edition

1. Specifications:
Describes system specifications of sample tasks.
2. Description of functions used:
Describes features and allocation of peripheral functions used in the sample tasks.
3. Description of operations:
Describes operations of sample tasks using timing charts.
4. Description of software:
 - Description of modules:
Describes software modules that run the sample tasks.
 - Description of arguments:
Describes input arguments needed when executing modules and output arguments after execution.
 - Description of internal registers:
Describes internal registers (timer control register, serial mode register etc.) of peripheral functions that are set by modules.
 - Description of general registers:
Describes general registers (R0–R7) used by modules, and their functions.
 - Description of RAM:
Covers label names and functions of RAM used by modules.
5. Flow charts:
Covers general flowcharts for software running the sample tasks.
6. Program list:
Covers program lists of software running the sample tasks.

1.2 Organization of Application Edition

The application edition is arranged as in figure 1.3, and describes methods of use for combinations of peripheral functions.

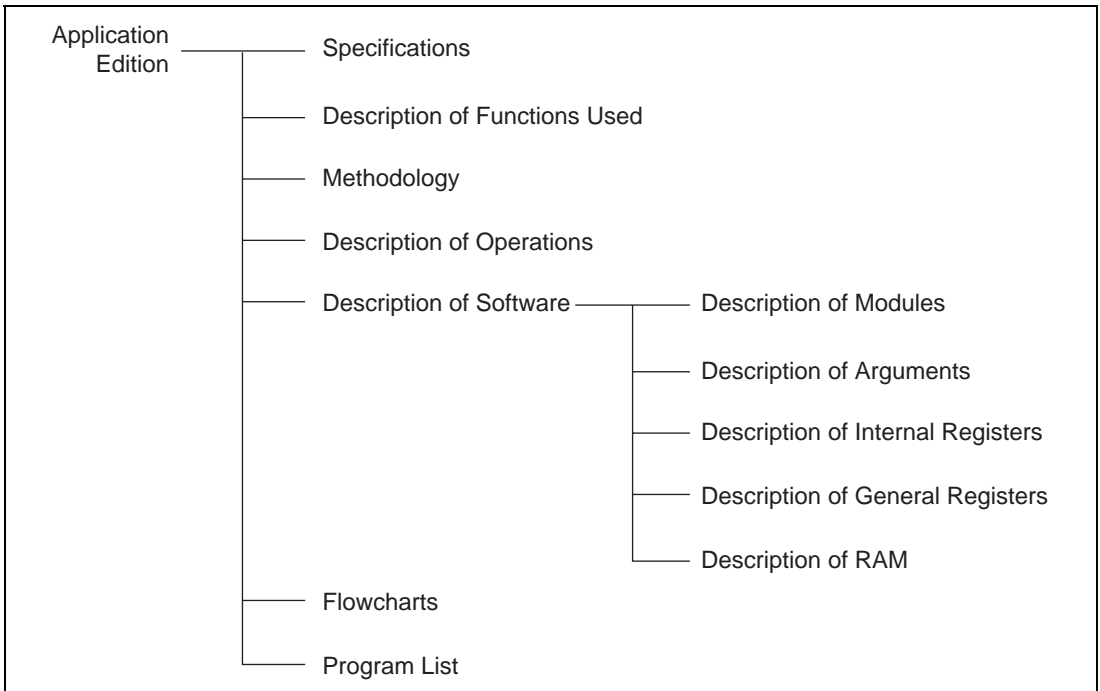


Figure 1.3 Organization of Application Edition

1. Specifications:
Describes system specifications of sample tasks.
2. Description of functions used:
Describes features and allocation of peripheral functions used in the sample tasks.
3. Method:
Describes methods of actualizing systems used in sample tasks.
4. Description of operations:
Describes operations of sample tasks using timing charts.
5. Description of software:
 - Description of modules:
Describes software modules that run the sample tasks.
 - Description of arguments:

Describes input arguments needed when executing modules and output arguments after execution.

— Description of internal registers:

Describes internal registers (timer control register, serial mode register etc.) of peripheral functions that are set by modules.

— Description of general registers:

Describes general registers (R0–R7) used by modules, and their functions.

— Description of RAM:

Describes label names and functions of RAM used by modules.

6. Flowcharts:

Covers general flowcharts for software running the sample tasks.

7. Program list:

Covers program lists of software executing the sample tasks.

2.1 Pulse Output

MCU: H8/3003

Function: ITU (16-bit integrated timer unit)

2.1.1 Specifications

Output of pulse with 50% duty of the cycle set in RAM (see figure 2.1). Output pulse cycle can be set optionally from 125 ns to 4.09 ms during 16 MHz operation.

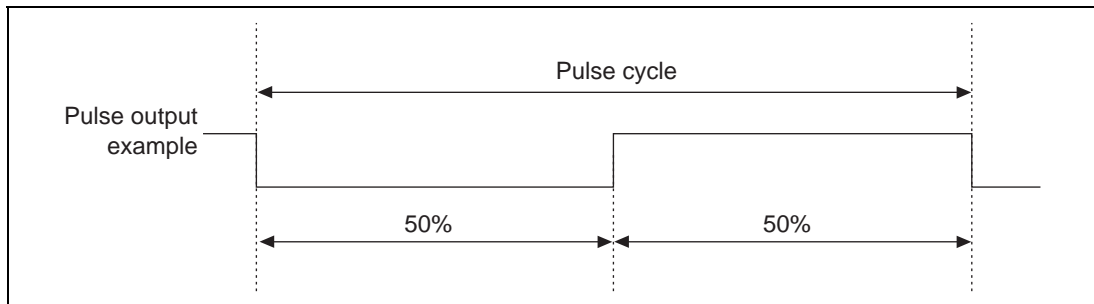


Figure 2.1 Pulse Output

2.1.2 Description of Functions Used

In this sample task, a 50% duty pulse is output using ITU/ch0. Figure 2.2 shows a block diagram of the ITU/ch0 used in this sample task.

The following functions are used in ch0:

- Automatic pulse output function without software intervention (output compare).
- Counter clear function during compare/match (counter clear).
- Output reversal with each compare/match (toggle output).

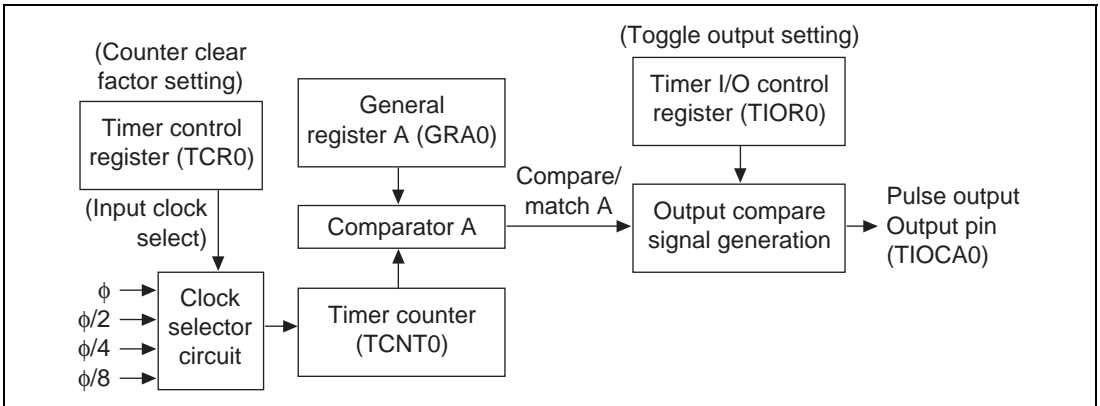


Figure 2.2 ITU/ch0 Block Diagram

Table 2.1 shows the function allocation of this sample task. ITU functions are allocated for pulse output.

Table 2.1 ITU Function Allocation

ITU function	Function
TCR0	Selects clock input to TCNT and counter clear factor
TIOCA0	Pulse output
TIOR0	Sets pulse output level
GRA0	Sets pulse 1/2 cycle

2.1.3 Description of Operations

Figure 2.3 shows the operation principle. Pulses are output due to H8/3003 hardware and software processing.

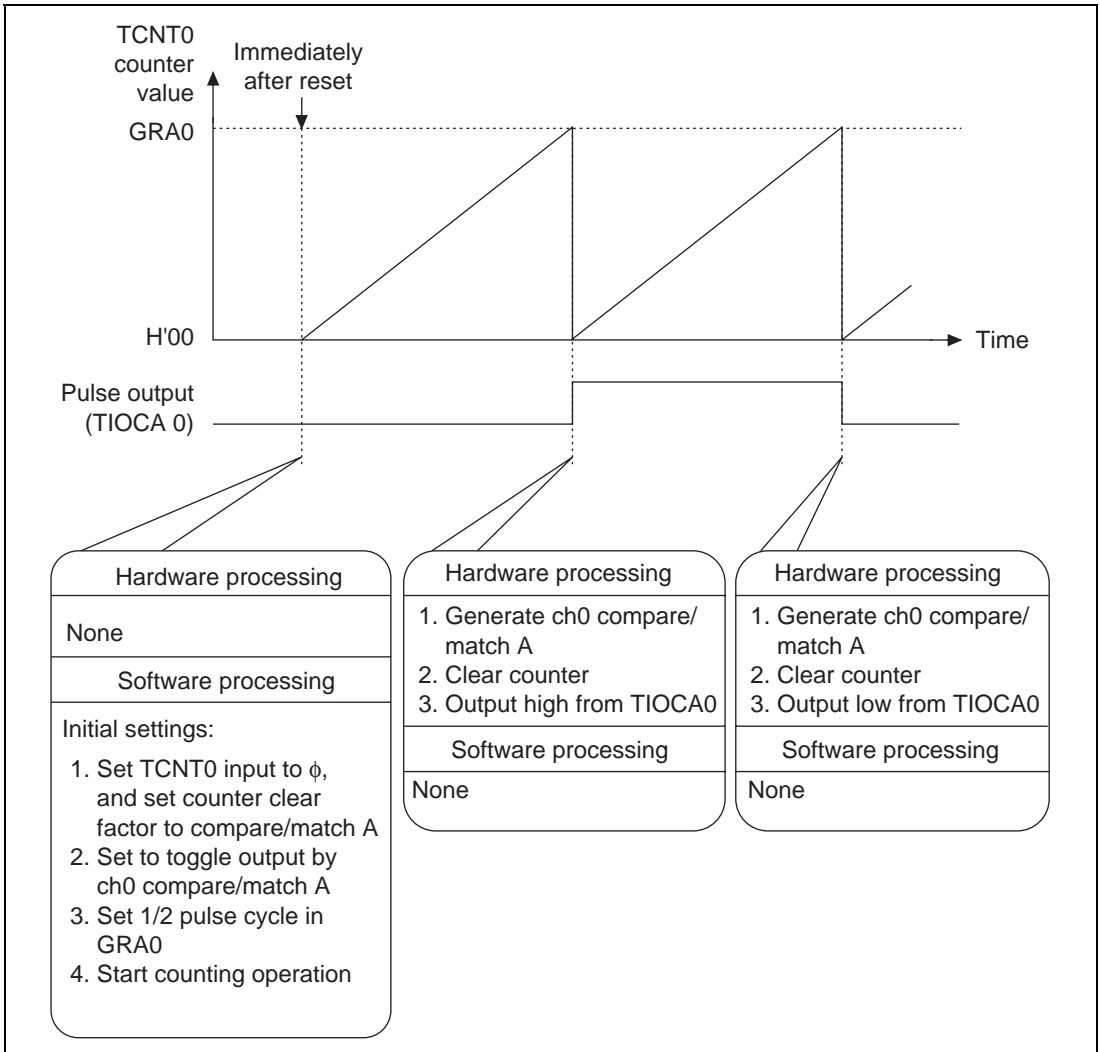


Figure 2.3 Principle of Pulse Output Operation

2.1.4 Description of Software

The software for pulse output is described below.

Table 2.2 Description of Modules

Module Name	Label Name	Function
Main routine	POUTMN	ITU and RAM initial setting and pulse output

Table 2.3 Description of Arguments

Label Name, Register Name	Function	Data length	Module Name	I/O
PUL_CYC	Sets timer value equivalent to pulse cycle The pulse cycle is determined by the following formula:Pulse cycle (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)	1 word	Main routine	Input

Table 2.4 Description of Internal Registers

Register Name	Function	Module Name	
TSTR	Timer counter operation enable/disable setting	Main routine	
ch0	TCR0	Sets clock input to TCNT and counter clear factor	Main routine
	TIOR0	Sets level of output pulse during compare/match A generation	Main routine
	GRA0	Sets output pulse 1/2 cycle	Main routine

Table 2.5 Description of General Registers

Module Name	Register Name	Function
Main routine	R0	Used as work register during data setting.

Description of RAM: RAM not used other than for arguments in this sample task.

2.1.5 Flowchart

Figure 2.4 shows the pulse output flowchart.

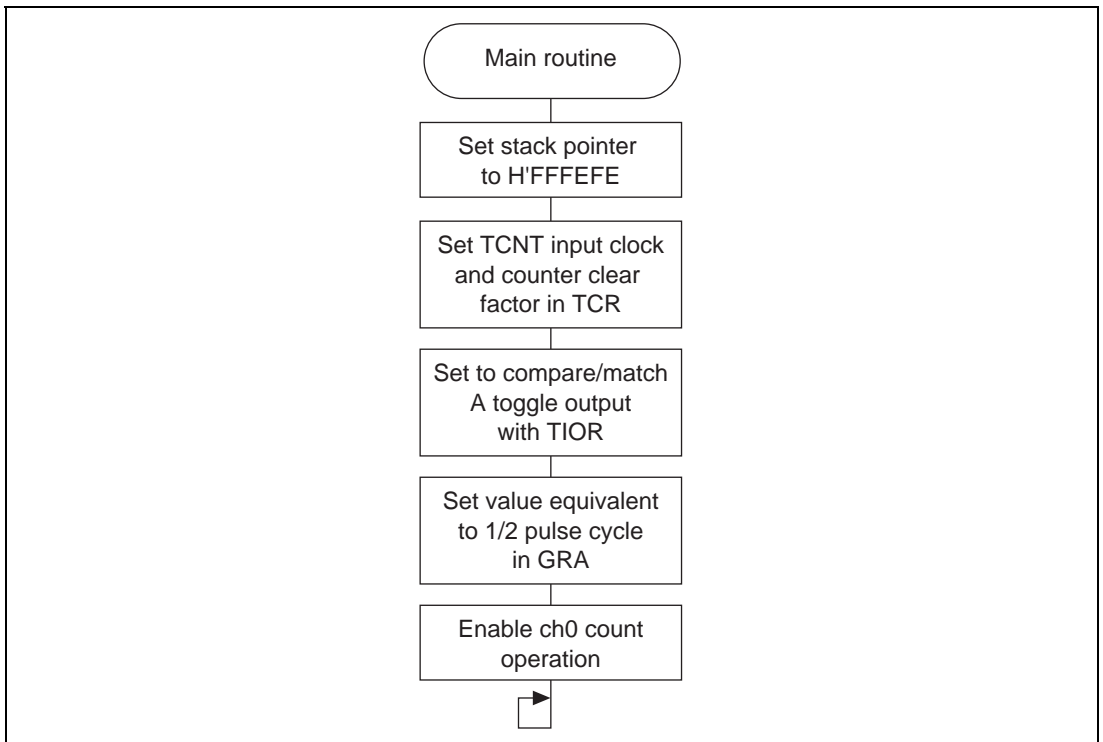


Figure 2.4 Main Routine Flowchart

2.1.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION  VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES      .DATA.L  POUTMN
10         10 ;
11 00001C   11         .ORG      H'00001C
12         12 ;
13 00001C 00010000  13 NMI      .DATA.L  POUTMN
14 000020 00010000  14 TRAPA0   .DATA.L  POUTMN
15 000024 00010000  15 TRAPA1   .DATA.L  POUTMN
16 000028 00010000  16 TRAPA2   .DATA.L  POUTMN
17 00002C 00010000  17 TRAPA3   .DATA.L  POUTMN
18         18 ;
19 000030   19         .ORG      H'000030
20         20 ;
21 000030 00010000  21 IRQ0     .DATA.L  POUTMN
22 000034 00010000  22 IRQ1     .DATA.L  POUTMN
23 000038 00010000  23 IRQ2     .DATA.L  POUTMN
24 00003C 00010000  24 IRQ3     .DATA.L  POUTMN
25 000040 00010000  25 IRQ4     .DATA.L  POUTMN
26 000044 00010000  26 IRQ5     .DATA.L  POUTMN
27 000048 00010000  27 IRQ6     .DATA.L  POUTMN
28 00004C 00010000  28 IRQ7     .DATA.L  POUTMN
29 000050 00010000  29 WOVI0    .DATA.L  POUTMN
30 000054 00010000  30 CMI      .DATA.L  POUTMN
31         31 ;
32 000060   32         .ORG      H'000060
33 000060 00010000  33 IMIA0    .DATA.L  POUTMN
34 000064 00010000  34 IMIB0    .DATA.L  POUTMN
35 000068 00010000  35 OVI0     .DATA.L  POUTMN
36         36 ;
37 000070   37         .ORG      H'000070
38 000070 00010000  38 IMIA1    .DATA.L  POUTMN
39 000074 00010000  39 IMIB1    .DATA.L  POUTMN
40 000078 00010000  40 OVI1     .DATA.L  POUTMN
41         41 ;
42 000080   42         .ORG      H'000080
43 000080 00010000  43 IMIA2    .DATA.L  POUTMN

```

```

44 000084 00010000      44 IMIB2      .DATA.L      POUTMN
45 000088 00010000      45 OVI2       .DATA.L      POUTMN
46                               46 ;
47 000090                               47           .ORG           H'000090
48 000090 00010000      48 IMIA3      .DATA.L      POUTMN
49 000094 00010000      49 IMIB3      .DATA.L      POUTMN
50 000098 00010000      50 OVI3       .DATA.L      POUTMN
51                               51 ;
52 0000A0                               52           .ORG           H'0000A0
53 0000A0 00010000      53 IMIA4      .DATA.L      POUTMN
54 0000A4 00010000      54 IMIB4      .DATA.L      POUTMN
55 0000A8 00010000      55 OVI4       .DATA.L      POUTMN
56                               56 ;
57 0000B0                               57           .ORG           H'0000B0
58 0000B0 00010000      58 DEND0A     .DATA.L      POUTMN
59 0000B4 00010000      59 DEND0B     .DATA.L      POUTMN
60 0000B8 00010000      60 DEND1A     .DATA.L      POUTMN
61 0000BC 00010000      61 DEND1B     .DATA.L      POUTMN
62 0000C0 00010000      62 DEND2A     .DATA.L      POUTMN
63 0000C4 00010000      63 DEND2B     .DATA.L      POUTMN
64 0000C8 00010000      64 DEND3A     .DATA.L      POUTMN
65 0000CC 00010000      65 DEND3B     .DATA.L      POUTMN
66 0000D0 00010000      66 ERI0       .DATA.L      POUTMN
67 0000D4 00010000      67 RXI0       .DATA.L      POUTMN
68 0000D8 00010000      68 TXI0       .DATA.L      POUTMN
69 0000DC 00010000      69 TEI0       .DATA.L      POUTMN
70 0000E0 00010000      70 ERI1       .DATA.L      POUTMN
71 0000E4 00010000      71 RXI1       .DATA.L      POUTMN
72 0000E8 00010000      72 TXI1       .DATA.L      POUTMN
73 0000EC 00010000      73 TEI1       .DATA.L      POUTMN
74 0000F0 00010000      74 ADI        .DATA.L      POUTMN
75                               75 ;
76                               76 ;*****
77                               77 ;* *
78                               78 ;*      RAM ALLOCATION *
79                               79 ;* *
80                               80 ;*****
81                               81 ;
82 FFFF00                               82           .SECTION   RAM,DATA,LOCATE=H'FFFF00
83                               83 ;
84 FFFF00 00000002      84 PUL_CYC    .RES.W      1           ;Pulse cycle time
85                               85 ;
86                               86 ;*****
87                               87 ;* *
88                               88 ;*      SYMBOL DEFINITIONS *
89                               89 ;* *

```

```

90 ;*****
91 ;
92 00FFFF60 92 TSTR .EQU H'FFFF60 ;Timer start register
93 ;
94 00FFFF64 94 TCR0 .EQU H'FFFF64 ;Timer control register
95 00FFFF65 95 TIOR0 .EQU H'FFFF65 ;Timer I/O control register
96 00FFFF6A 96 GRA0 .EQU H'FFFF6A ;General register A0
97 ;
98 ;*****
99 ;* *
100 ;* MAIN PROGRAM : POUTMN *
101 ;* *
102 ;*****
103 ;
104 010000 104 .SECTION PROG, CODE, LOCATE=H'010000
105 ;
106 00010000 106 POUTMN: .EQU $
107 010000 7A0700FFFEFE 107 MOV.L #H'FFFEFE, SP ;Initialize stack pointer
108 010006 F8A0 108 MOV.B #B'10100000, R0L
109 010008 3864 109 MOV.B R0L, @TCR0 ;Initialize TCR0
110 01000A F88B 110 MOV.B #B'10001011, R0L
111 01000C 3865 111 MOV.B R0L, @TIOR0 ;Initialize TIOR0
112 ;
113 01000E 6B00FF00 113 MOV.W @PUL_CYC, R0
114 010012 1110 114 SHLR.W R0
115 010014 6B80FF6A 115 MOV.W R0, @GRA0
116 ;
117 010018 F8E1 117 MOV.B #B'11100001, R0L
118 01001A 3860 118 MOV.B R0L, @TSTR ;Set TCNT0 start data
119 ;
120 01001C 40FE 120 POUTMN99 BRA POUTMN99
121 ;
122 .END

****TOTAL ERRORS 0
****TOTAL WARNINGS 0

```

2.2 Two-Phase Encoder Count

MCU: H8/3003

Function: ITU (phase calculation mode)

2.2.1 Specifications

As shown in figure 2.5, two external clocks are input, and the counter counts up or down due to the phase difference between the two inputs. Also, the number of counts is taken and the result set in RAM.

With the timer counter's initial value at H'8000, counting can be conducted up to +7FFF and -8000.

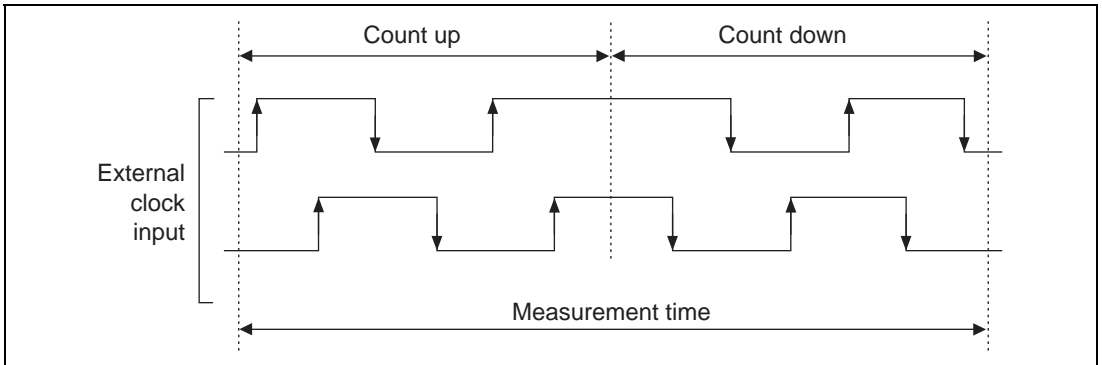


Figure 2.5 Two-Phase Encoder Count Operation Example

2.2.2 Description of Functions Used

In this sample task, ITU ch2 is used for the up/down counter, and the measurement time is generated in ch0. The ch0 output compare output pin and ch2 input capture pin are connected, and notification of measurement time end is given to ch2 due to ch0 output compare.

Figure 2.6 shows a block diagram of ch0. ch0 generates measurement time using the following function, providing pulse output to the ch2 input capture pin.

- Automatic pulse output function in hardware without software intervention (output compare).

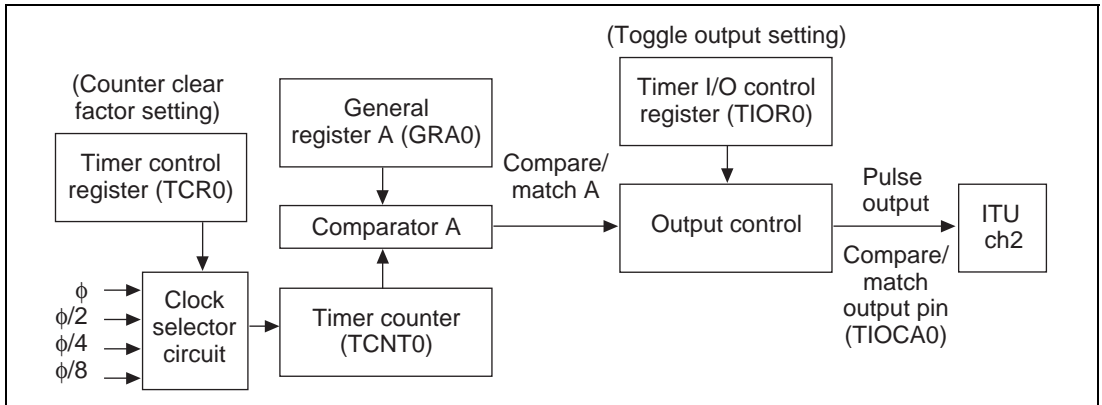


Figure 2.6 ITU/ch0 Block Diagram

Figure 2.7 shows a block diagram of ch2. ch2 uses the following functions for up/down counting the timer counter. The counter value at the time of input capture both-edge detection is taken as the measurement result.

- Phase difference detection of two external clocks for timer counter up/down counting (phase calculation mode).
- Setting of timer value in internal register at the time of pulse input edge detection (input capture).
- Startup of interrupt processing at time of input capture generation.
- Startup of interrupt processing at time of timer counter overflow or underflow detection.

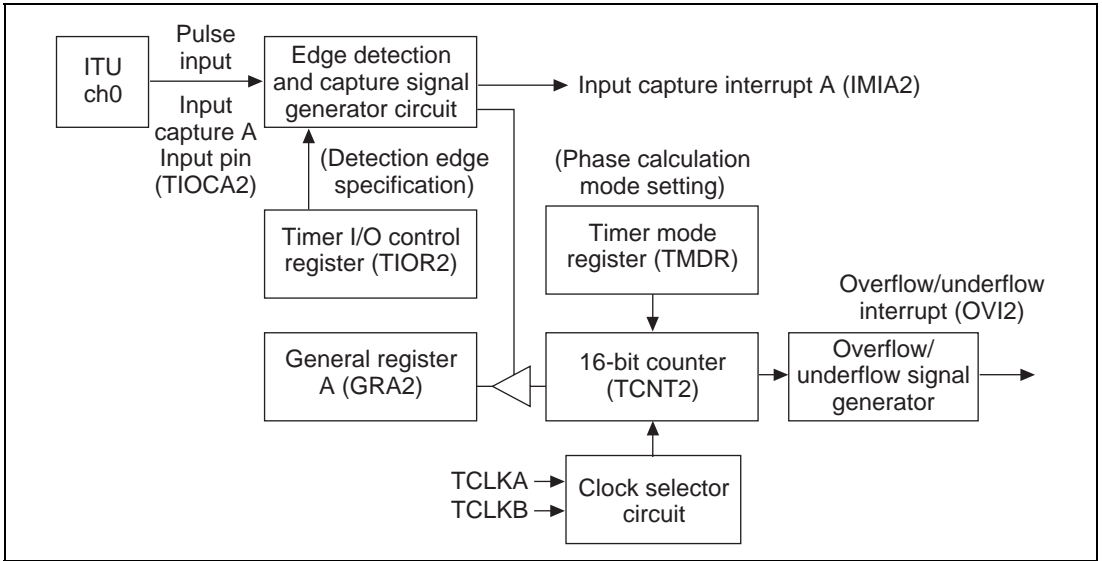


Figure 2.7 ITU/ch2 Block Diagram

Table 2.6 shows the function allocation for this sample task. ITU functions are allocated for two-phase encoder pulse phase difference detection and counter up/down counting.

Table 2.6 ITU Function Allocation

ITU Function		Function
TSTR		Enables/disables ch0/ch2 timer counter operation
TMDR		Sets phase calculation mode and OVR flag set condition
TCLKA		External clock input pin
TCLKB		External clock input pin
ch0	TCR0	Selects counter clock and counter clear factor
	TIOCA0	Outputs pulse
	TIOR0	Sets toggle output with compare/match A
	GRA0	Sets counter measurement time
ch2	TIOCA2	Input pin for input capture signal
	TIER2	Enables interrupts due to IMFA and OVF
	TIOR2	Sets to input capture A generation during edge detection
	GRA2	Counter result set due to input capture A

2.2.3 Description of Operation

Figure 2.8 shows the operation principle. The counter is counted up/down due to H8/3003 hardware and software processing.

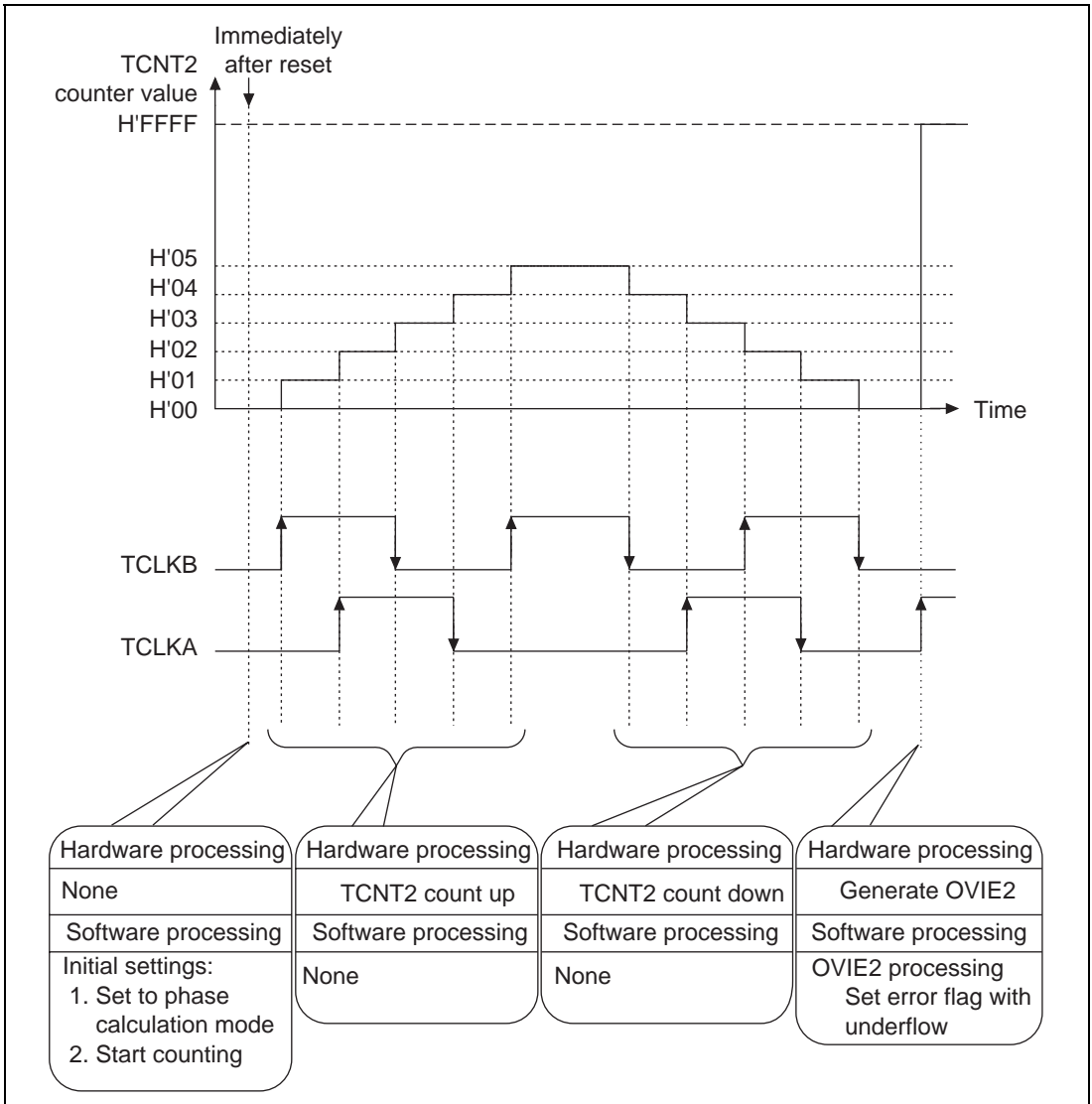


Figure 2.8 Operation Principle of Phase Calculation Mode (1)

Figure 2.8 shows the H8/3003 hardware and software processing for interrupt processing during overflow/underflow and external event generation.

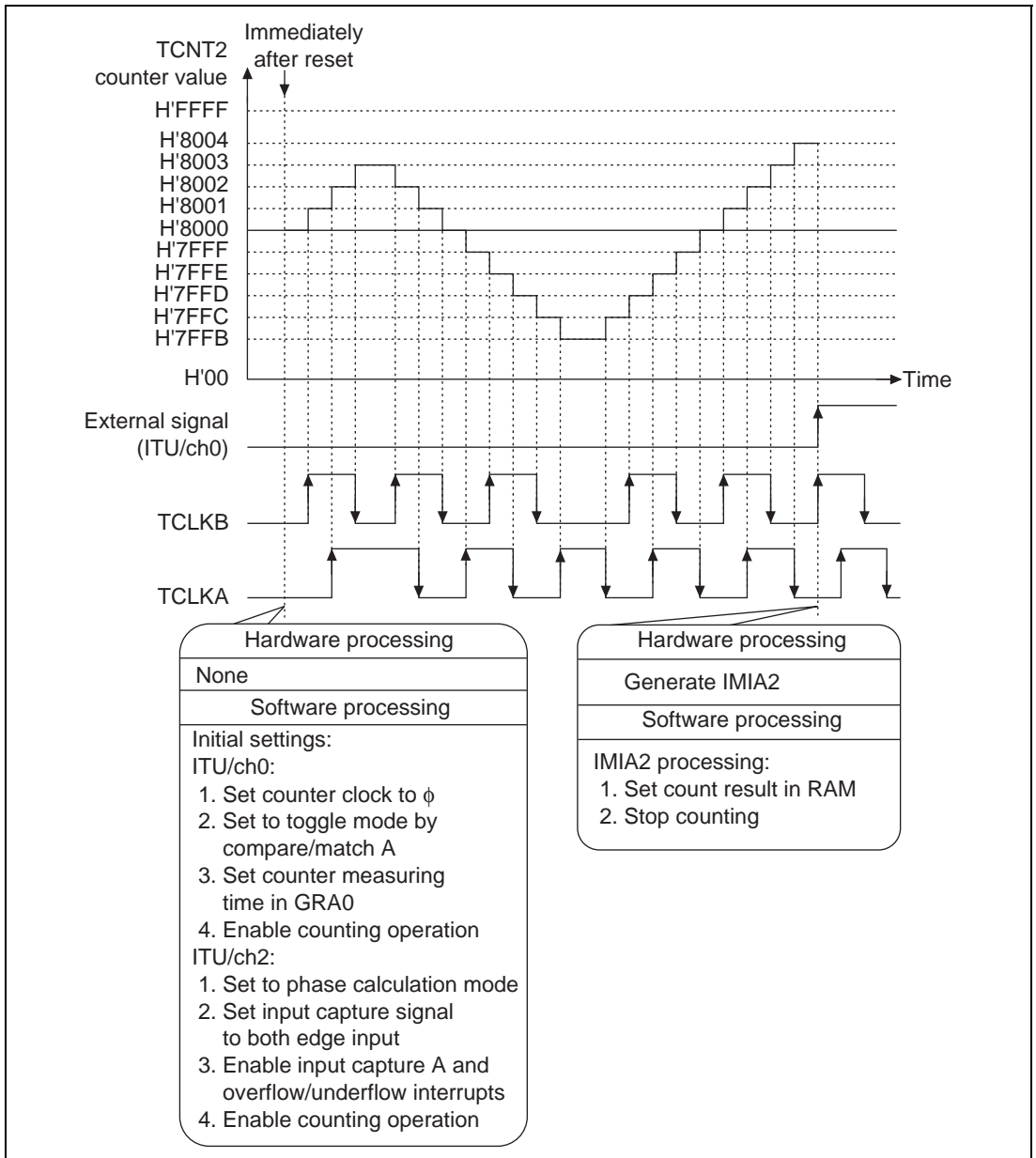


Figure 2.8 Operation Principle of Phase Calculation Mode (2)

2.2.4 Description of Software

The software for phase calculation mode is described below.

Table 2.7 Description of Modules

Module name	Label name	Function
Main routine	PHACNTMN	Initial setting of ITU and RAM
Counter value measurement	PHACNT1	Startup due to IMIA2, and setting of up/down count result from GRA in RAM
Error flag setting	PHACNT2	Startup due to OVIE2, and setting of error flag by overflow/underflow

Table 2.8 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
MSR_TIM	Sets timer value equivalent to counter measurement time. Measurement time is determined by the following formula: Measurement time (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)	1 word	Main routine	I
CNT_DATA	Sets up/down count result	1 word	Counter value measurement	O
ERR_F	Error flag setting due to overflow/underflow	1 bit	Error flag setting	O

Table 2.9 Description of Internal Registers

Module Name	Register Name	Function	
TSTS	Timer counter operation enable/disable setting	Main routine	
TMDR	Sets phase calculation mode and OVF flag set condition		
ch0	TCR0	Selects TCNT counter clock and counter clear factor	
	TIOR0	Sets output compare to toggle output	
	GRA0	Sets counter measurement time	
ch2	TIOR2	Selects input capture A generation by external signal falling edge	
	GRA2	Sets counter value during input capture A	Counter value measurement
	TIER2	Enables interrupts due to IMFA and OVF	Main routine
	TSR2	Indicates occurrence of interrupts due to IMFA and OVF	Error flag setting

Table 2.10 Description of General Registers

Module Name	Register Name	Function
Main routine	R0L	Used as work register during data setting
Counter value measurement	R0	Used as work register during data setting

Table 2.11 Description of RAM

Label Name, Register Name	Function	Data Length	Module Name
CNT_OVR	Stores error flag	1 byte	Error flag setting

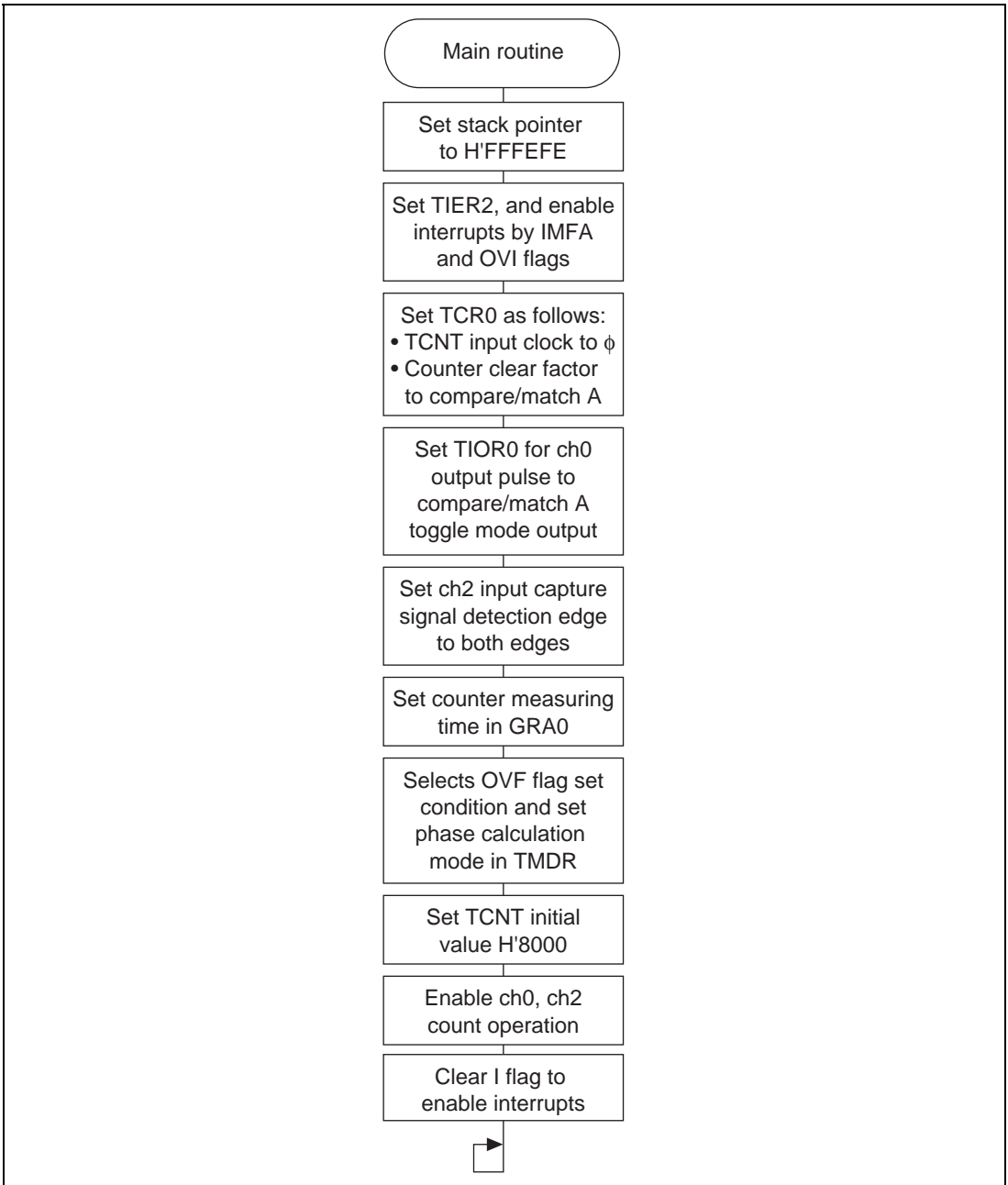


Figure 2.9 Main Routine Flowchart

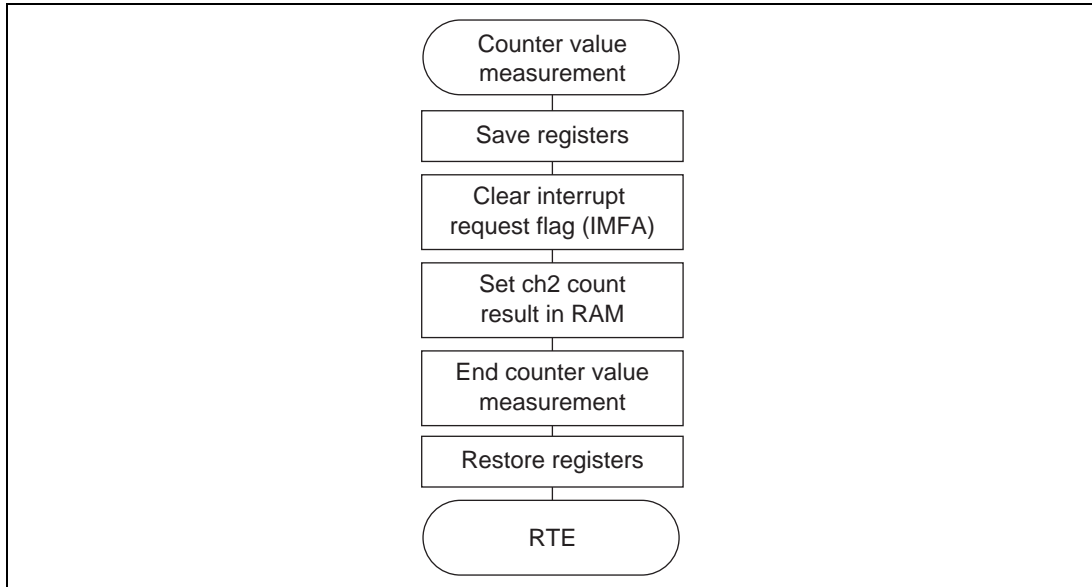


Figure 2.10 Counter Value Measurement Flowchart

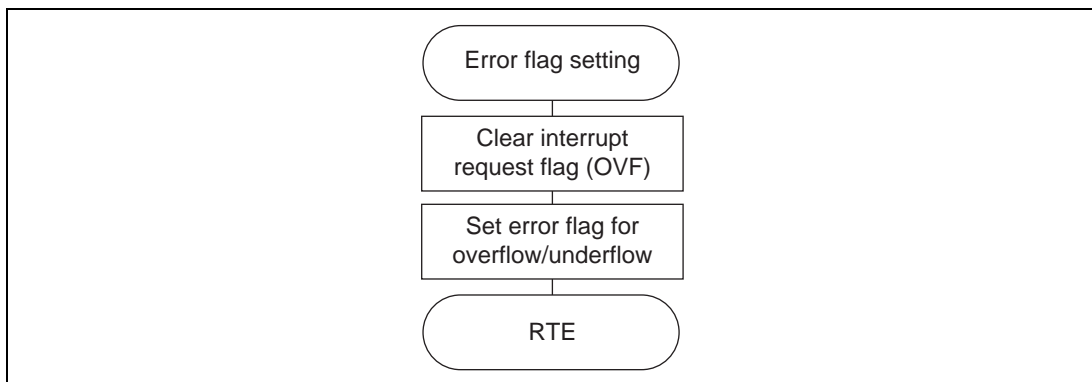


Figure 2.11 Error Flag Setting Flowchart

2.2.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION  VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000   9 RES          .DATA.L  PHACNTMN
10         10 ;
11 00001C  11          .ORG      H'00001C
12 00001C   12 NMI        .DATA.L  PHACNTMN
13         13 ;
14 000030  14          .ORG      H'000030
15 000030   15 IRQ0       .DATA.L  PHACNTMN
16 000034   16 IRQ1       .DATA.L  PHACNTMN
17 000038   17 IRQ2       .DATA.L  PHACNTMN
18 00003C   18 IRQ3       .DATA.L  PHACNTMN
19 000040   19 IRQ4       .DATA.L  PHACNTMN
20 000044   20 IRQ5       .DATA.L  PHACNTMN
21 000048   21 IRQ6       .DATA.L  PHACNTMN
22 00004C   22 IRQ7       .DATA.L  PHACNTMN
23 000050   23 WOVI0     .DATA.L  PHACNTMN
24 000054   24 CMI        .DATA.L  PHACNTMN
25         25 ;
26 000060  26          .ORG      H'000060
27 000060   27 IMIA0      .DATA.L  PHACNTMN
28 000064   28 IMIB0      .DATA.L  PHACNTMN
29 000068   29 OVI0       .DATA.L  PHACNTMN
30         30 ;
31 000070  31          .ORG      H'000070
32 000070   32 IMIA1      .DATA.L  PHACNTMN
33 000074   33 IMIB1      .DATA.L  PHACNTMN
34 000078   34 OVI1       .DATA.L  PHACNTMN
35         35 ;
36 000080  36          .ORG      H'000080
37 000080   37 IMIA2      .DATA.L  PHACNT1
38 000084   38 IMIB2      .DATA.L  PHACNTMN
39 000088   39 OVI2       .DATA.L  PHACNT2
40         40 ;
41 000090  41          .ORG      H'000090
42 000090   42 IMIA3      .DATA.L  PHACNTMN
43 000094   43 IMIB3      .DATA.L  PHACNTMN

```

```

44 000098 00010000      44 OVI3      .DATA.L      PHACNTMN
45                               45 ;
46 0000A0                               46          .ORG      H'0000A0
47 0000A0 00010000      47 IMIA4      .DATA.L      PHACNTMN
48 0000A4 00010000      48 IMIB4      .DATA.L      PHACNTMN
49 0000A8 00010000      49 OVI4      .DATA.L      PHACNTMN
50                               50 ;
51 0000B0                               51          .ORG      H'0000B0
52 0000B0 00010000      52 DEND0A     .DATA.L      PHACNTMN
53 0000B4 00010000      53 DEND0B     .DATA.L      PHACNTMN
54 0000B8 00010000      54 DEND1A     .DATA.L      PHACNTMN
55 0000BC 00010000      55 DEND1B     .DATA.L      PHACNTMN
56 0000C0 00010000      56 DEND2A     .DATA.L      PHACNTMN
57 0000C4 00010000      57 DEND2B     .DATA.L      PHACNTMN
58 0000C8 00010000      58 DEND3A     .DATA.L      PHACNTMN
59 0000CC 00010000      59 DEND3B     .DATA.L      PHACNTMN
60 0000D0 00010000      60 ERI0      .DATA.L      PHACNTMN
61 0000D4 00010000      61 RXI0      .DATA.L      PHACNTMN
62 0000D8 00010000      62 TXI0      .DATA.L      PHACNTMN
63 0000DC 00010000      63 TEI0      .DATA.L      PHACNTMN
64 0000E0 00010000      64 ERI1      .DATA.L      PHACNTMN
65 0000E4 00010000      65 RXI1      .DATA.L      PHACNTMN
66 0000E8 00010000      66 TXI1      .DATA.L      PHACNTMN
67 0000EC 00010000      67 TEI1      .DATA.L      PHACNTMN
68 0000F0 00010000      68 ADI      .DATA.L      PHACNTMN
69                               69 ;
70                               70 ;*****
71                               71 ;* *
72                               72 ;*      RAM ALLOCATION *
73                               73 ;* *
74                               74 ;*****
75                               75 ;
76 FFFF00                               76          .SECTION  RAM,DATA,LOCATE=H'FFFF00
77                               77 ;
78 FFFF00 00000002      78 MSR_TIM   .RES.W      1          ;Measure time
79 FFFF02 00000002      79 CNT_DATA  .RES.W      1          ;count result
80 FFFF04 00000001      80 CNT_OVR   .RES.B      1
81          00000000      81 ERR_F     .EQU       0          ;Error flag
82                               82 ;
83                               83 ;*****
84                               84 ;* *
85                               85 ;*      SYMBOL DEFINITIONS *
86                               86 ;* *
87                               87 ;*****
88                               88 ;
89          00FFFF60      89 TSTR     .EQU       H'FFFF60      ;Timer start register

```

```

90      00FFFF62      90 TMDR      .EQU      H'FFFF62      ;Timer mode register
91
92      00FFFF64      92 TCR0      .EQU      H'FFFF64      ;Timer control register0
93      00FFFF65      93 TIOR0     .EQU      H'FFFF65      ;Timer I/O control register0
94      00FFFF6A      94 GRA0      .EQU      H'FFFF6A      ;General register 0A
95
96      00FFFF79      96 TIOR2     .EQU      H'FFFF79      ;Timer I/O control register2
97      00FFFF7A      97 TIER2     .EQU      H'FFFF7A      ;Timer interrupt enable
                      ;register2
98      00FFFF7B      98 TSR2      .EQU      H'FFFF7B      ;Timer status register2
99      00000000      99 IMPA2     .EQU      0          ;Inputcapture flag2
100     00000002      100 OVF2     .EQU      2          ;Over/Under flow flag2
101     00FFFF7C      101 TCNT2    .EQU      H'FFFF7C      ;Timer counter 2
102     00FFFF7E      102 GRA2     .EQU      H'FFFF7E      ;General register A2
103
104
105
106
107
108
109
110     010000      110          .SECTION  PROG,CODE,LOCATE=H'010000
111
112     00010000      112 PHACNTMN: .EQU      $
113     010000      7A0700FFFEFE 113          MOV.L      #H'FFFEFE,SP ;Initialize stack pointer
114
115     010006      F8FD        115          MOV.B      #B'11111101,R0L
116     010008      387A        116          MOV.B      R0L,@TIER2 ;Initialize TIER0
                      ;(1->IMIEA)
117     01000A      F8A0        117          MOV.B      #B'10100000,R0L
118     01000C      3864        118          MOV.B      R0L,@TCR0 ;Initialize TCR0
119     01000E      F88B        119          MOV.B      #B'10001011,R0L
120     010010      3865        120          MOV.B      R0L,@TIOR0 ;Initialize TIOR0
121     010012      6B00FF00    121          MOV.W      @MSR_TIM,R0
122     010016      6B80FF6A    122          MOV.W      R0,@GRA0
123
124     01001A      F88F        124          MOV.B      #B'10001111,R0L
125     01001C      3879        125          MOV.B      R0L,@TIOR2 ;Initialize TIOR2
126
127     01001E      F8C0        127          MOV.B      #B'11000000,R0L
128     010020      3862        128          MOV.B      R0L,@TMDR ;Set phase counting mode
129
130     010022      79008000    130          MOV.W      #H'8000,R0
131     010026      6B80FF7C    131          MOV.W      R0,@TCNT2 ;Set Initialize data
132     01002A      F8E5        132          MOV.B      #B'11100101,R0L
133     01002C      3860        133          MOV.B      R0L,@TSTR ;Start TCNT0,2

```

```

134
135 01002E 0700          135          LDC.B      #0,CCR      ;Enable interrupt
136 010030 40FE          136 PHACNTMN99  BRA      PHACNTMN99
137
138
139
140
141
142
143
144
145
146
147
148
149          00010032      149 PHACNT1   .EQU      $
150 010032 6DF0          150          PUSH.W    R0          ;Escape register
151 010034 7F7B7200      151          BCLR      #IMFA2,@TSR2 ;Clear IMFA0 request
                                ;(0->IMFA0)
152 010038 6B00FF7E      152          MOV.W     #GRA2,R0
153 01003C 6B80FF02      153          MOV.W     R0,@CNT_DATA ;Store count data
154 010040 F8F8          154          MOV.B     #B'11111000,R0L
155 010042 387A          155          MOV.B     R0L,@TIER2
156 010044 F8E0          156          MOV.B     #B'11100000,R0L
157 010046 3860          157          MOV.B     R0L,@TSTR     ;Stop counter
158 010048 6D70          158          POP.W     R0          ;Return register
159 01004A 5670          159          RTE
160
161
162
163
164
165
166
167
168
169
170
171
172          0001004C      172 PHACNT2   .EQU      $
173 01004C 7F7B7220      173          BCLR      #OVF2,@TSR2 ;Clear OVF2 request
                                ;(0->OVF2)
174 010050 7F047000      174          BSET     #ERR_F,@CNT_OVR ;Set error flag
175 010054 5670          175          RTE
176
177          .END

```

*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0

2.3 Pulse High and Low Width Measurement

MCU: H8/3003

Function: ITU

2.3.1 Specifications

As shown in figure 2.12, pulse high and low width times are measured and the results stored in RAM.

During 16 MHz operation, pulse high and low widths can be measured in units of 62.5 ns, from 5.2 microsec to 4.08 ms.

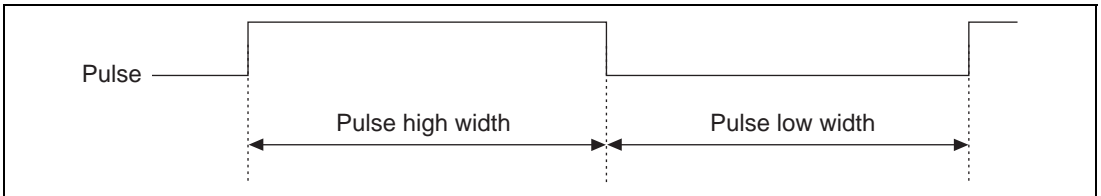


Figure 2.12 Pulse Width Measurement Timing

2.3.2 Description of Functions Used

This sample task measures pulse high and low widths using ch0.

Figure 2.13 shows a block diagram of ch0. This sample task uses the following functions:

- Pulse rising and falling edge detection, and setting of the timer value at these times in an internal register (input capture)
- Timer counter clearing at time of input capture
- Startup of interrupt processing at time of pulse rising edge and falling edge detection

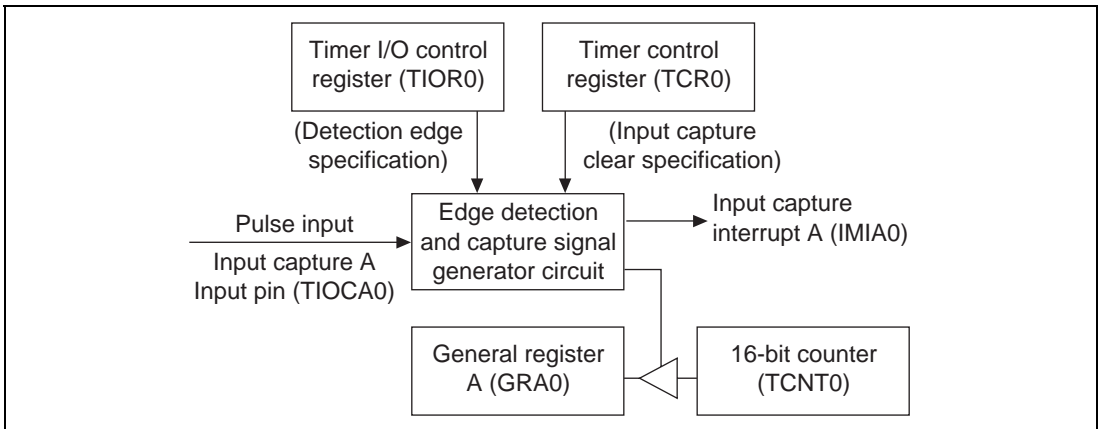


Figure 2.13 ITU/ch0 Block Diagram

Table 2.12 shows the function allocation for this sample task. ITU functions are allocated for measurement of pulse high and low widths.

Table 2.12 ITU Function Allocation

ITU Function		Function
ch0	TCR0	Selects counter clear factor
	TIOR0	Selects input edge of input capture signal
	TIOCA0	Inputs pulse for measurement
	GRA0	Detects counter value at time of pulse rising/falling edge
	IMIA0	Starts up pulse high and low width measurement due to pulse rising/falling edge

2.3.3 Description of Operations

Figure 2.14 shows the principle of operation. Pulse high and low widths are measured due to the H8/3003's hardware and software processing.

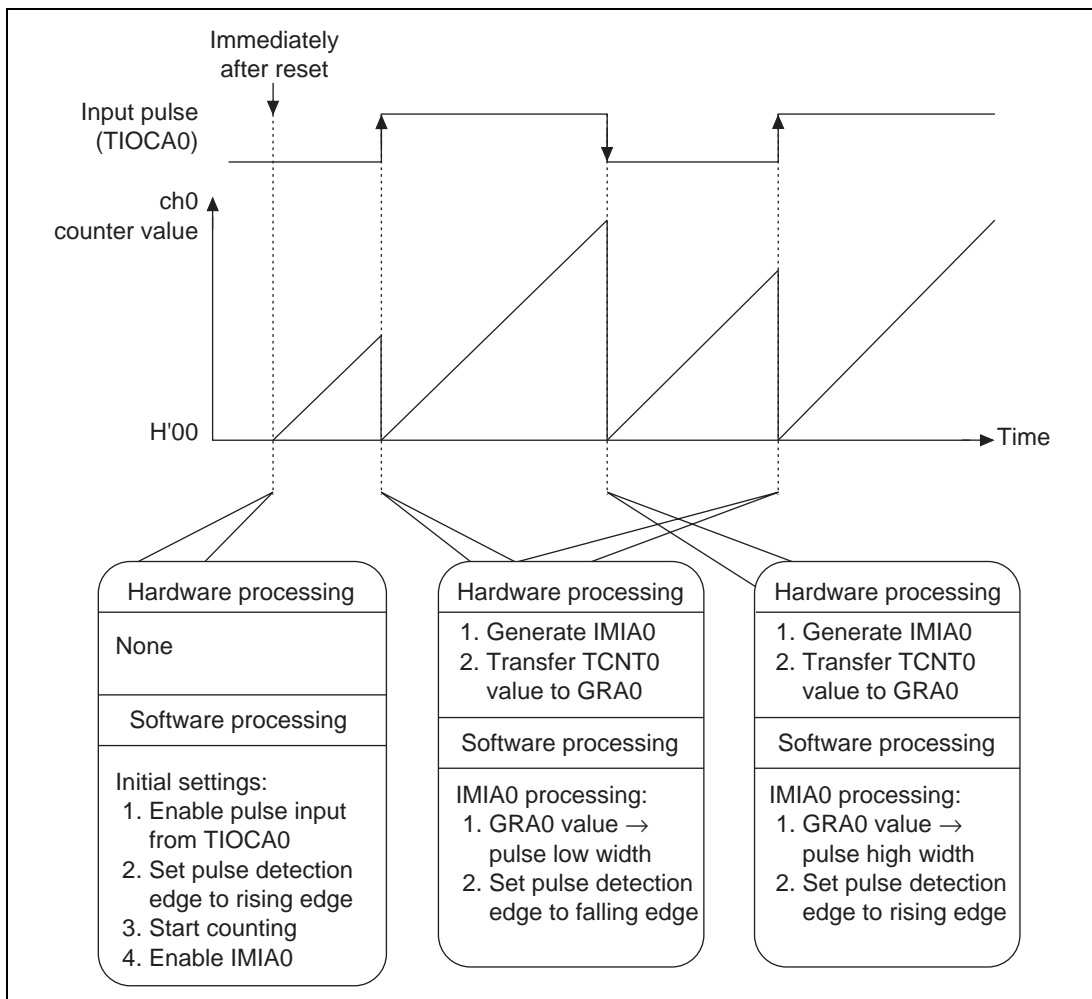


Figure 2.14 Operation Principle of Pulse Width Measurement

2.3.4 Description of Software

The software for pulse high and low width measurement is described below.

Table 2.13 Description of Modules

Module Name	Label Name	Function
Main routine	PWHLMN	Initial setting of ITU and RAM
Pulse high and low width measurement	PWHL1	Startup by IMIA, measurement of pulse high and low widths from GRA value, and setting in RAM

Table 2.14 Description of Arguments

Label Name, Register Name	Function	Data length	Module Name	I/O
PWH_HDATA	Setting timer value equivalent to pulse high width. Pulse high width is determined from the following formula:Pulse high width (ns) = timer value \times ϕ cycle(62.5 ns during 16 MHz operation)	1 word	Pulse high and low width measurement	Output
PWH_LDATA	Setting timer value equivalent to pulse low width. Pulse low width is determined from the following formula:Pulse low width (ns) = timer value \times ϕ cycle(62.5 ns during 16 MHz operation)	1 word		

Table 2.15 Description of Internal Registers

Register Name	Function	Module Name
TSTR	Timer counter operation enable/disable setting	Main routine
ch0	TCR0	Selects TCNT counter clock, and sets counter clear factor to input capture A
	TIOR0	Sets for transfer from TCNT to GRA due to pulse rising edge/falling edge detection
	TIER0	Enables interrupts with IMFA flag
	GRA0	Sets TCNT value at time of pulse rising edge/falling edge, and determines pulse cycle from this value
	TSR0	Indicates input capture A generation

Table 2.16 Description of General Registers

Module Name	Register Name	Function
Main routine	R0	Used as work register during data setting
Pulse high and low width measurement	R0	Used as work register during data setting

Description of RAM: RAM not used other than for arguments in this sample task.

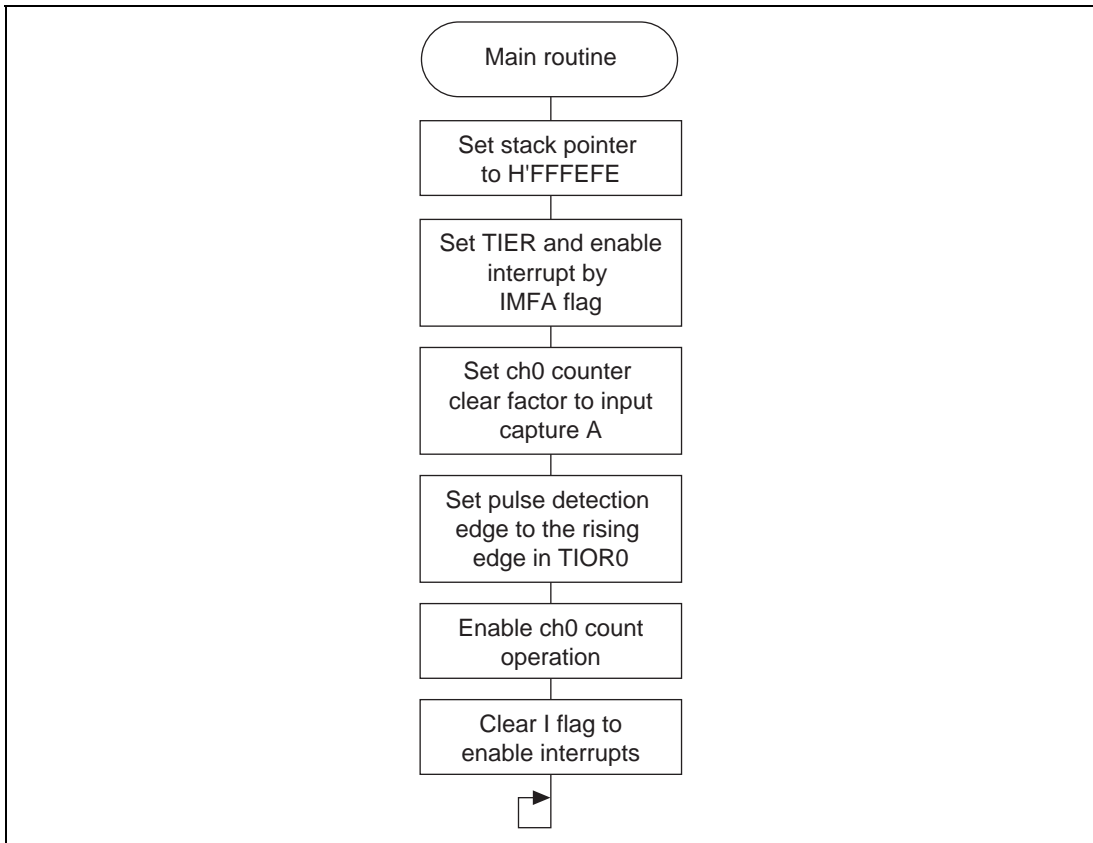


Figure 2.15 Main Routine Flowchart

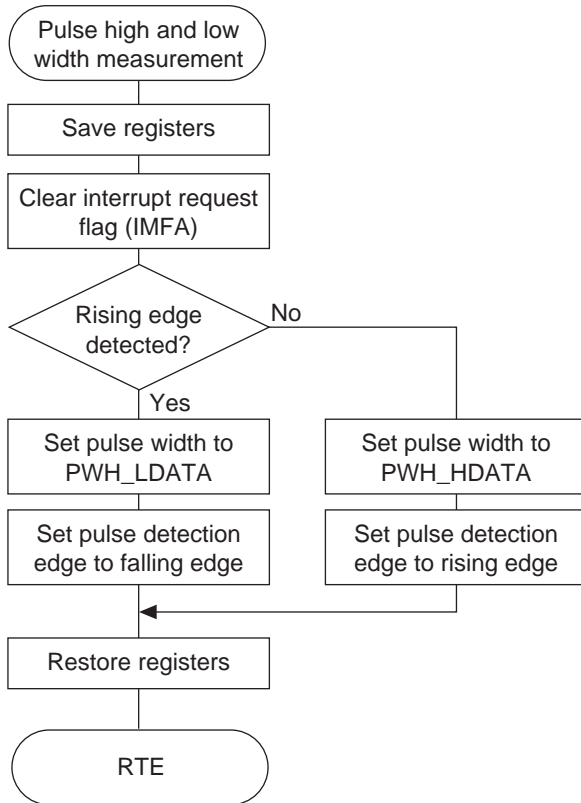


Figure 2.16 Pulse High and Low Width Measurement Flowchart

2.3.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION  VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000   9 RES          .DATA.L  PWHLMN
10         10 ;
11 00001C  11          .ORG      H'00001C
12 00001C  12 NMI          .DATA.L  PWHLMN
13 000020  13 TRAPA0       .DATA.L  PWHLMN
14 000024  14 TRAPA1       .DATA.L  PWHLMN
15 000028  15 TRAPA2       .DATA.L  PWHLMN
16 00002C  16 TRAPA3       .DATA.L  PWHLMN
17         17 ;
18 000030  18          .ORG      H'000030
19 000030  19 IRQ0           .DATA.L  PWHLMN
20 000034  20 IRQ1           .DATA.L  PWHLMN
21 000038  21 IRQ2           .DATA.L  PWHLMN
22 00003C  22 IRQ3           .DATA.L  PWHLMN
23 000040  23 IRQ4           .DATA.L  PWHLMN
24 000044  24 IRQ5           .DATA.L  PWHLMN
25 000048  25 IRQ6           .DATA.L  PWHLMN
26 00004C  26 IRQ7           .DATA.L  PWHLMN
27 000050  27 WOVI0         .DATA.L  PWHLMN
28 000054  28 CMI           .DATA.L  PWHLMN
29         29 ;
30 000060  30          .ORG      H'000060
31 000060  31 IMIA0         .DATA.L  PWHL1
32 000064  32 IMIB0         .DATA.L  PWHLMN
33 000068  33 OVI0         .DATA.L  PWHLMN
34         34 ;
35 000070  35          .ORG      H'000070
36 000070  36 IMIA1         .DATA.L  PWHLMN
37 000074  37 IMIB1         .DATA.L  PWHLMN
38 000078  38 OVI1         .DATA.L  PWHLMN
39         39 ;
40 000080  40          .ORG      H'000080
41 000080  41 IMIA2         .DATA.L  PWHLMN
42 000084  42 IMIB2         .DATA.L  PWHLMN
43 000088  43 OVI2         .DATA.L  PWHLMN

```

```

44 ;
45 000090 45 .ORG H'000090
46 000090 00010000 46 IMIA3 .DATA.L PWHLMN
47 000094 00010000 47 IMIB3 .DATA.L PWHLMN
48 000098 00010000 48 OVI3 .DATA.L PWHLMN
49 ;
50 0000A0 50 .ORG H'0000A0
51 0000A0 00010000 51 IMIA4 .DATA.L PWHLMN
52 0000A4 00010000 52 IMIB4 .DATA.L PWHLMN
53 0000A8 00010000 53 OVI4 .DATA.L PWHLMN
54 ;
55 0000B0 55 .ORG H'0000B0
56 0000B0 00010000 56 DEND0A .DATA.L PWHLMN
57 0000B4 00010000 57 DEND0B .DATA.L PWHLMN
58 0000B8 00010000 58 DEND1A .DATA.L PWHLMN
59 0000BC 00010000 59 DEND1B .DATA.L PWHLMN
60 0000C0 00010000 60 DEND2A .DATA.L PWHLMN
61 0000C4 00010000 61 DEND2B .DATA.L PWHLMN
62 0000C8 00010000 62 DEND3A .DATA.L PWHLMN
63 0000CC 00010000 63 DEND3B .DATA.L PWHLMN
64 0000D0 00010000 64 ERI0 .DATA.L PWHLMN
65 0000D4 00010000 65 RXI0 .DATA.L PWHLMN
66 0000D8 00010000 66 TXI0 .DATA.L PWHLMN
67 0000DC 00010000 67 TEI0 .DATA.L PWHLMN
68 0000E0 00010000 68 ERI1 .DATA.L PWHLMN
69 0000E4 00010000 69 RXI1 .DATA.L PWHLMN
70 0000E8 00010000 70 TXI1 .DATA.L PWHLMN
71 0000EC 00010000 71 TEI1 .DATA.L PWHLMN
72 0000F0 00010000 72 ADI .DATA.L PWHLMN
73 ;
74 ;*****
75 ;* *
76 ;* RAM ALLOCATION *
77 ;* *
78 ;*****
79 ;
80 FFFF00 80 .SECTION RAM,DATA,LOCATE=H'FFFF00
81 ;
82 FFFF00 00000002 82 PWH_HDATA .RES.W 1 ;Pulse high width time
83 FFFF02 00000002 83 PWH_LDATA .RES.W 1 ;Pulse low width time
84 ;
85 ;*****
86 ;* *
87 ;* SYMBOL DEFINITIONS *
88 ;* *
89 ;*****

```

```

90          90 ;
91          00FFFF60      91 TSTR      .EQU      H'FFFF60      ;Timer start register
92          92 ;
93          00FFFF64      93 TCR0      .EQU      H'FFFF64      ;Timer control register0
94          00FFFF65      94 TIOR0     .EQU      H'FFFF65      ;Timer I/O control register0
95          00000000      95 IOA0      .EQU      0          ;I/O control A0
96          00FFFF66      96 TIER0     .EQU      H'FFFF66      ;Timer interrupt enable
                                     ;register0
97          00FFFF67      97 TSR0      .EQU      H'FFFF67      ;Timer status register0
98          00000000      98 IMFA0     .EQU      0          ;Inputcapture flag A0
99          00FFFF6A      99 GRA0      .EQU      H'FFFF6A      ;General register A0
100         100 ;
101         101 ;*****
102         102 ;*
103         103 ;*      MAIN PROGRAM : PWHLMN      *
104         104 ;*
105         105 ;*****
106         106 ;
107         010000      107          .SECTION  PROG,CODE,LOCATE=H'010000
108         108 ;
109         00010000      109 PWHLMN:   .EQU      $
110         010000      7A0700FFFEFE 110          MOV.L      #H'FFFEFE.SP ;Initialize stack pointer
111         111 ;
112         010006      F8F9      112          MOV.B      #B'11111001,R0L
113         010008      3866      113          MOV.B      R0L,@TIER0 ;Initialize TIER0
                                     ;(1->IMIEA)
114         01000A      F8A0      114          MOV.B      #B'10100000,R0L
115         01000C      3864      115          MOV.B      R0L,@TCR0 ;Initialize TCR0
116         116 ;
117         01000E      F88C      117          MOV.B      #B'10001100,R0L
118         010010      3865      118          MOV.B      R0L,@TIOR0 ;Initialize TIOR0
119         119 ;
120         010012      F8E1      120          MOV.B      #B'11100001,R0L
121         010014      3860      121          MOV.B      R0L,@TSTR ;Start TCNT0
122         010016      0700      122          LDC.B      #0,CCR ;Enable interrupt
123         123
124         010018      40FE      124 PWHLMN99 BRA      PWHLMN99
125         125 ;
126         126 ;*****
127         127 ;*
128         128 ;*      NAME : PWHL1(MEASUREMENT PULSE H AND L)      *
129         129 ;*
130         130 ;*****
131         131 ;*
132         132 ;*      ENTRY : NOTHING
133         133 ;*      RETURNS : PWH_HDATA(PULSE HIGH PERIOD DATA)      *

```

```

134          134 ;* RETURNS : PWH_LDATAL(PULSE LOW PERIOD DATA) *
135          135 ;* *
136          136 ;*****
137          137 ;
138          138 PWHLL1 .EQU $
139 01001A 6DF0 139 PUSH.W R0 ;Escape register
140 01001C 7F677200 140 BCLR #IMFA0,@TSR0 ;Clear IMFA0 request
; (0->IMFA0)
141 010020 6B00FF6A 141 MOV.W @GRA0,R0 ;Load now data
142 010024 7E657300 142 BTST #IOA0,@TIOR0 ;High period
; measurement end?
143 010028 58600008 143 BNE PWHLL100 ;Yes
144 144 ;
145 01002C 6B80FF02 145 MOV.W R0,@PWH_LDATAL ;No, store low data
146 010030 58000004 146 BRA PWHLL101
147 147 ;
148 010034 6B80FF00 148 PWHLL100 MOV.W R0,@PWH_HDATA ;Yes, store high data
149 149 ;
150 010038 7F657100 150 PWHLL101 BNOT #IOA0,@TIOR0 ;Invert IOA0
151 151 ;
152 01003C 6D70 152 POP.W R0 ;Return register
153 01003E 5670 153 RTE
154 154
155 155
156 156 .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0

```


2.4 Complementary PWM Three-Phase Output

MCU: H8/3003

Function: ITU (complementary PWM mode)

2.4.1 Specifications

3-phase output of PWM waveforms with a non-overlap relation of positive and negative phases (see figure 2.17).

Duty can be set optionally from 0% to 100%.

$$\text{Duty} = \text{Pulse high width} / \text{Pulse cycle} \times 100 (\%)$$

Switching from ITU output pin to normal port due to external signal.

During 16 MHz operation, the output pulse cycle can be optionally set from 375 ns to 8.19 ms.

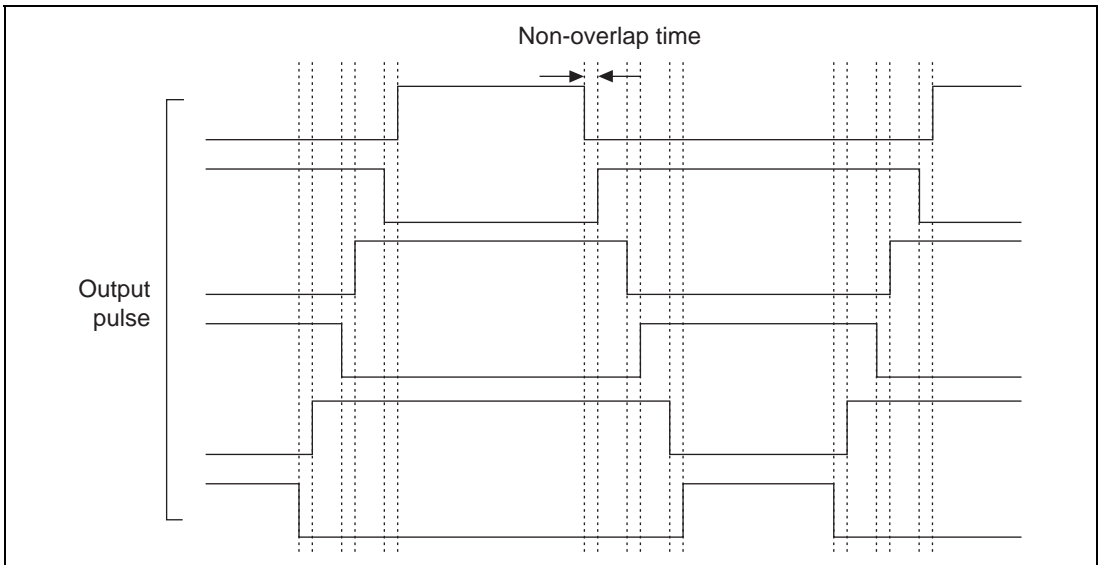


Figure 2.17 Positive/Negative Phase PWM 3-Phase Output Waveforms

2.4.2 Description of Functions Used

In this sample task, ch3 and ch4 are used to output PWM waveforms in three phases with a non-overlap relation of positive and negative phases. ch3 and ch4 output settings are prohibited due to ch1 input capture.

Figure 2.18 shows a block diagram of ITU/ch3/ch4 used in this sample task. The sample task uses the following functions.

- 3-phase output of PWM waveforms with a non-overlap relation of positive and negative phases (complementary PWM mode).
- Switching from ch3 and ch4 ITU output pins to normal port due to ch1 input capture signal.

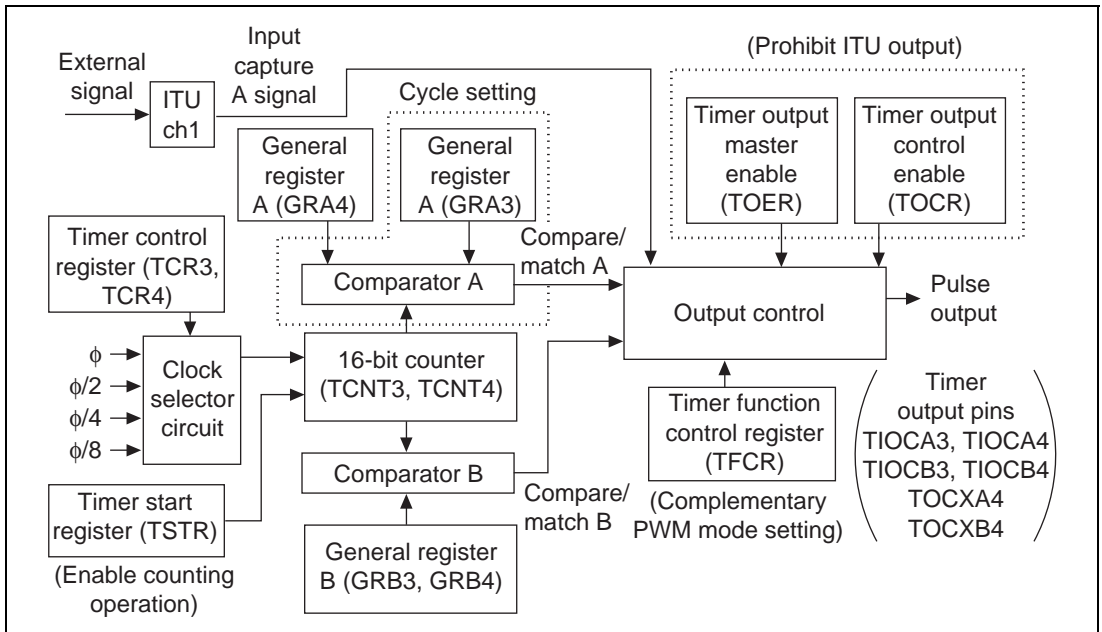


Figure 2.18 ITU/ch3, 4 Block Diagram

Table 2.17 shows the function allocation for this sample task. ITU functions are allocated, and channels 3 and 4 are used for 3-phase output of PWM waveforms with a non-overlap relation of positive and negative phases.

Table 2.17 ITU Function Allocation

ITU Function		Function
TSTR		Enables/disables TCNT count operation
TFCR		Sets to reset sync PWM mode
TOER		Enables/disables ch3/ch4 output setting
TOCR		Prohibits complementary PWM mode output due to external trigger
ch3	TCR3	Selects clock input to TCNT
	TIOCA3	Outputs positive phase pulse
	TIOCB3	Outputs negative phase pulse
	GRA3	Sets output pulse 1/2 cycle
ch4	TCR4	Selects clock input to TCNT
	TIOCA4	Outputs positive phase pulse
	TOCXA4	
	TIOCB4	Outputs negative phase pulse
	TOCXB4	
	GRA4	Sets variation point of pulses output from TIOCA and TOCXA pins
	GRB4	Sets variation point of pulses output from TIOCB and TOCXB pins

2.4.3 Description of Operations

Figure 2.19 shows the principle of operation. Complementary PWM waveform output is achieved through H8/3003 hardware and software processing.

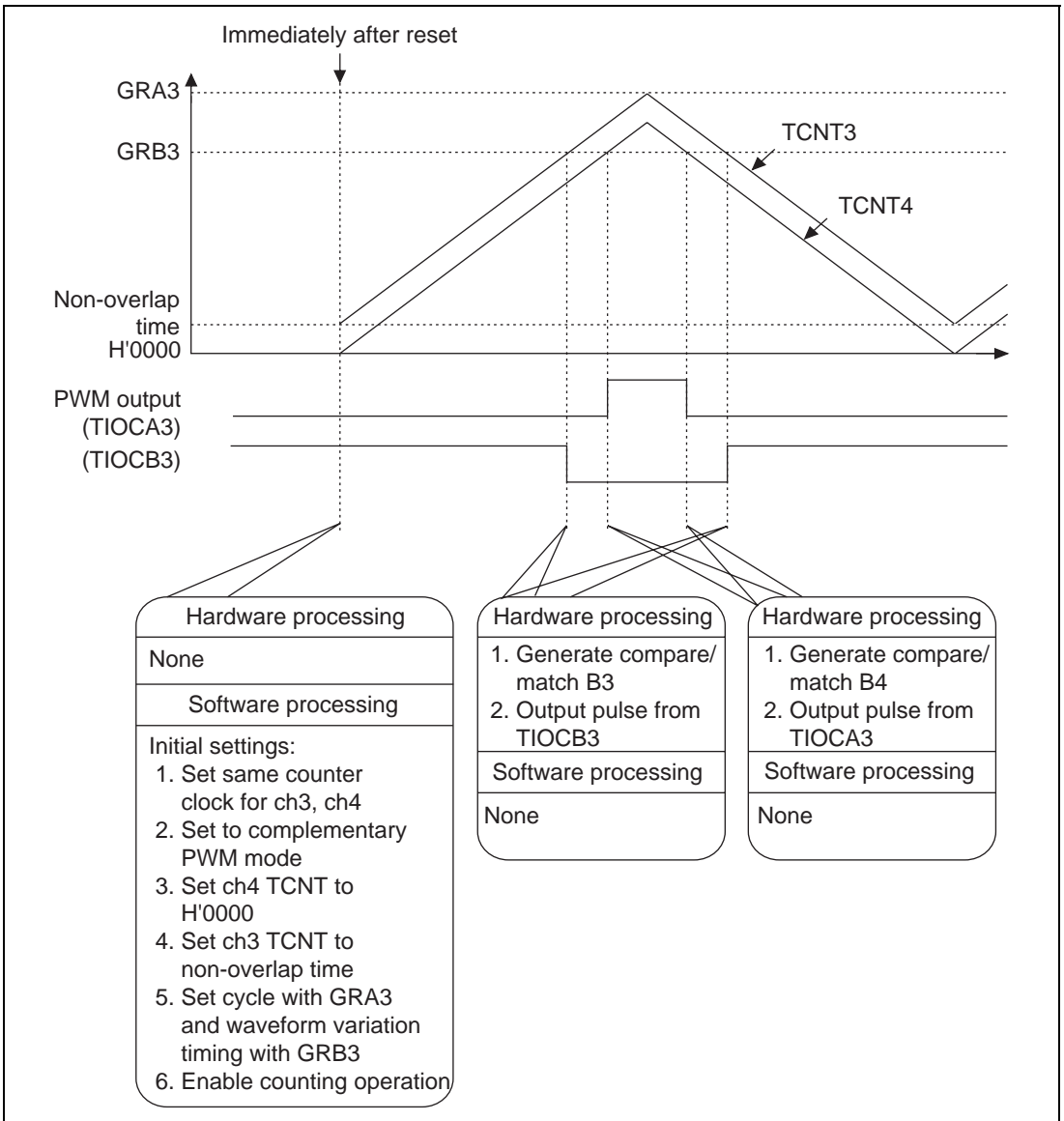


Figure 2.19 Operation Principle of Complementary PWM 1 Phase Output

Figure 2.20 shows the principle of operation. Complementary PWM waveform 3-phase output is achieved through H8/3003 hardware and software processing.

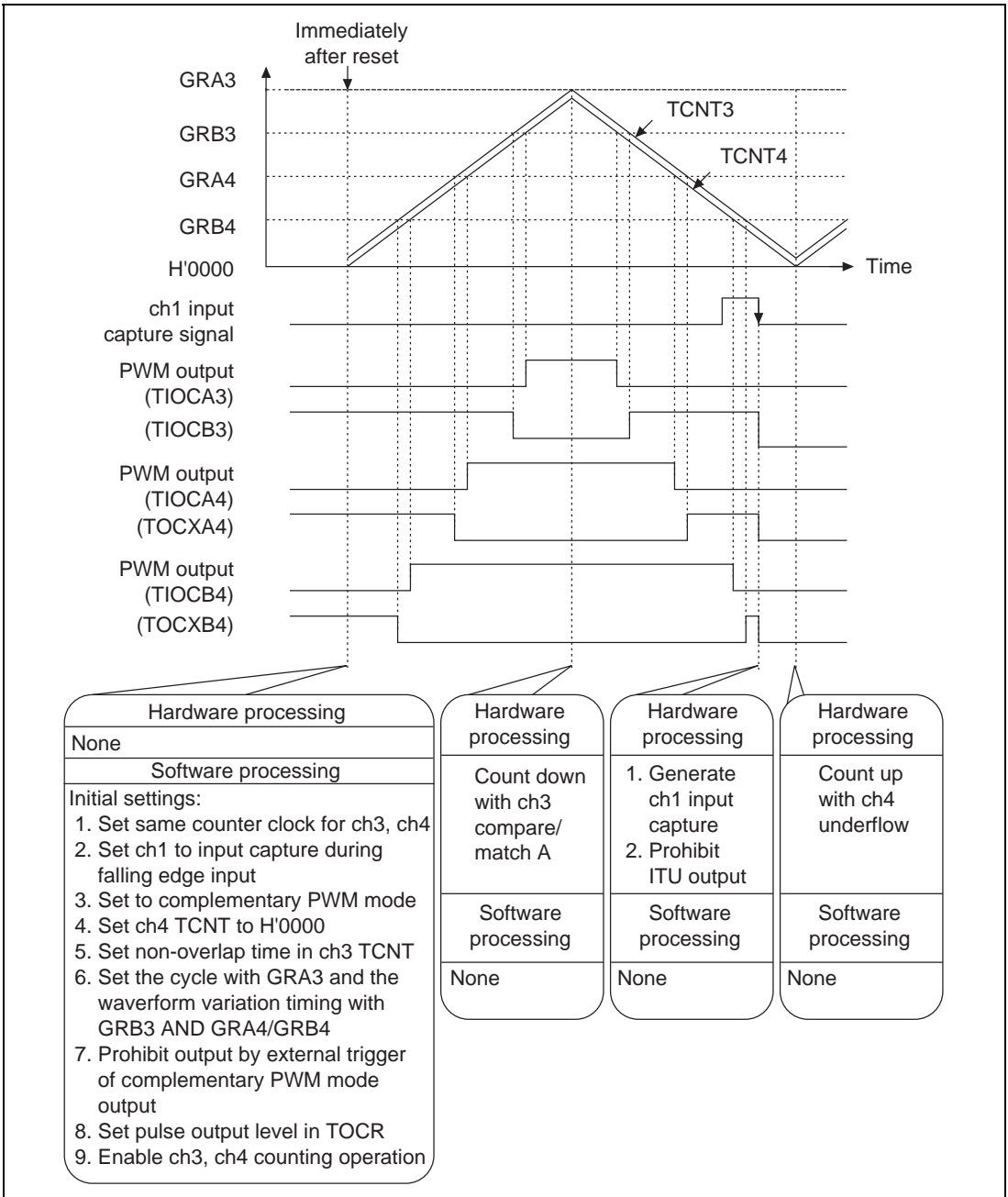


Figure 2.20 Operation Principle of Complementary PWM 3-Phase Output

2.4.4 Description of Software

The software for complementary PWM three-phase output is described below.

Table 2.18 Description of Modules

Module Name	Label Name	Function
Main routine	CMPWM3MN	Mode setting, setting of output pulse cycle and waveform variation timing, and initial setting of RAM

Table 2.19 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
PUL_CYC	Setting of timer value equivalent to 1/2 cycle of PWM pulse. Cycle is determined from the following formula: Cycle (ns) = timer value \times ϕ cycle (62.5 ns during 16 MHz operation) \leq upper limit value - 1 of TCNT3	1 word	Main routine	Input
PUL_TIMB3 PUL_TIMA4 PUL_TIMB4	Setting of timer value equivalent to waveform variation timing of PWM pulse. Waveform variation timing is determined by the following formula: Waveform variation timing (ns) = timer value \times ϕ cycle (62.5 ns during 16 MHz operation) provided TCNT3 and TCNT4 are set in the compare/match range.	1 word	Main routine	Input
NON_OVR	Setting of non-overlap time.	1 word	Main routine	Input

Table 2.20 Description of Internal Registers

Register Name		Function	Module Name
TSTR		Enables/disables timer counter operation	Main routine
TFCR		Sets ch3/ch4 to complementary PWM mode	Main routine
TOER		Enables/disables ch3/ch4 output setting	Main routine
TOCR		Prohibits output due to complementary PWM mode's external trigger	Main routine
ch1	TIOR1	Selects input capture A generation due to pulse falling edge input	Main routine
ch3	TCNT3	Sets non-overlap time	Main routine
	TCR3	Selects clock input to TCNT	Main routine
	GRA3	Sets output pulse 1/2 cycle	Main routine
	GRB3	Sets PWM output waveform variation timing	Main routine
ch4	TCNT4	Sets initial value H'0000	Main routine
	TCR4	Selects clock input to TCNT	Main routine
	GRA4	Sets PWM output waveform variation timing	Main routine
	GRB4	Sets PWM output waveform variation timing	Main routine

Table 2.21 Description of General Registers

Module Name	Register Name	Function
Main routine	R0	Used as work register during data setting

Description of RAM: RAM not used other than for arguments in this sample task.

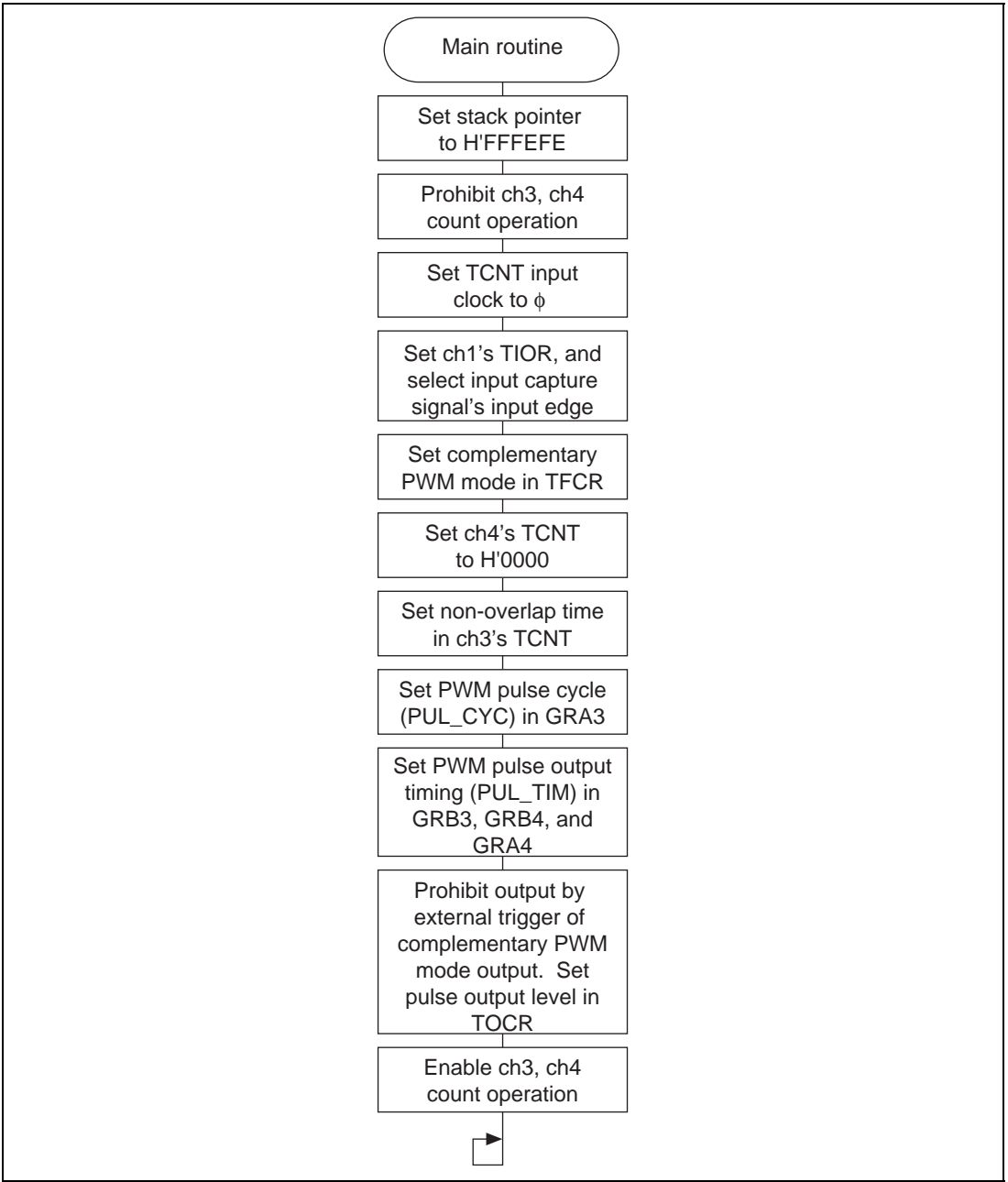


Figure 2.21 Main Routine Flowchart

2.4.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION  VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000   9 RES          .DATA.L  CMPWM3MN
10         10 ;
11 00001C  11          .ORG      H'00001C
12 00001C  12 NMI          .DATA.L  CMPWM3MN
13 000020  13 TRAPA0       .DATA.L  CMPWM3MN
14 000024  14 TRAPA1       .DATA.L  CMPWM3MN
15 000028  15 TRAPA2       .DATA.L  CMPWM3MN
16 00002C  16 TRAPA3       .DATA.L  CMPWM3MN
17         17 ;
18 000030  18          .ORG      H'000030
19 000030  19 IRQ0          .DATA.L  CMPWM3MN
20 000034  20 IRQ1          .DATA.L  CMPWM3MN
21 000038  21 IRQ2          .DATA.L  CMPWM3MN
22 00003C  22 IRQ3          .DATA.L  CMPWM3MN
23 000040  23 IRQ4          .DATA.L  CMPWM3MN
24 000044  24 IRQ5          .DATA.L  CMPWM3MN
25 000048  25 IRQ6          .DATA.L  CMPWM3MN
26 00004C  26 IRQ7          .DATA.L  CMPWM3MN
27 000050  27 WOVI0        .DATA.L  CMPWM3MN
28 000054  28 CMI          .DATA.L  CMPWM3MN
29         29 ;
30 000060  30          .ORG      H'000060
31 000060  31 IMIA0        .DATA.L  CMPWM3MN
32 000064  32 IMIB0        .DATA.L  CMPWM3MN
33 000068  33 OVI0        .DATA.L  CMPWM3MN
34         34 ;
35 000070  35          .ORG      H'000070
36 000070  36 IMIA1        .DATA.L  CMPWM3MN
37 000074  37 IMIB1        .DATA.L  CMPWM3MN
38 000078  38 OVI1        .DATA.L  CMPWM3MN
39         39 ;
40 000080  40          .ORG      H'000080
41 000080  41 IMIA2        .DATA.L  CMPWM3MN
42 000084  42 IMIB2        .DATA.L  CMPWM3MN
43 000088  43 OVI2        .DATA.L  CMPWM3MN

```

```

44 ;
45 000090 45 .ORG H'000090
46 000090 00010000 46 IMIA3 .DATA.L CMPWM3MN
47 000094 00010000 47 IMIB3 .DATA.L CMPWM3MN
48 000098 00010000 48 OVI3 .DATA.L CMPWM3MN
49 ;
50 0000A0 50 .ORG H'0000A0
51 0000A0 00010000 51 IMIA4 .DATA.L CMPWM3MN
52 0000A4 00010000 52 IMIB4 .DATA.L CMPWM3MN
53 0000A6 00010000 53 OVI4 .DATA.L CMPWM3MN
54 ;
55 0000B0 55 .ORG H'0000B0
56 0000B0 00010000 56 DEND0A .DATA.L CMPWM3MN
57 0000B4 00010000 57 DEND0B .DATA.L CMPWM3MN
58 0000B8 00010000 58 DEND1A .DATA.L CMPWM3MN
59 0000BC 00010000 59 DEND1B .DATA.L CMPWM3MN
60 0000C0 00010000 60 DEND2A .DATA.L CMPWM3MN
61 0000C4 00010000 61 DEND2B .DATA.L CMPWM3MN
62 0000C8 00010000 62 DEND3A .DATA.L CMPWM3MN
63 0000CC 00010000 63 DEND3B .DATA.L CMPWM3MN
64 0000D0 00010000 64 ERI0 .DATA.L CMPWM3MN
65 0000D4 00010000 65 RXI0 .DATA.L CMPWM3MN
66 0000D8 00010000 66 TXI0 .DATA.L CMPWM3MN
67 0000DC 00010000 67 TEI0 .DATA.L CMPWM3MN
68 0000E0 00010000 68 ERI1 .DATA.L CMPWM3MN
69 0000E4 00010000 69 RXI1 .DATA.L CMPWM3MN
70 0000E8 00010000 70 TXI1 .DATA.L CMPWM3MN
71 0000EC 00010000 71 TEI1 .DATA.L CMPWM3MN
72 0000F0 00010000 72 ADI .DATA.L CMPWM3MN
73 ;
74 ;*****
75 ;* *
76 ;* RAM ALLOCATION *
77 ;* *
78 ;*****
79 ;
80 FFFF00 80 .SECTION RAM,DATA,LOCATE=H'FFFF00
81 ;
82 FFFF00 00000002 82 NON_OVR .RES.W 1 ;Non-overlap time
83 ;
84 FFFF02 00000002 84 PUL_CYC .RES.W 1 ;Pulse cycle time
85 ;
86 FFFF04 00000002 86 PUL_TIMB3 .RES.W 1 ;Pulse set/reset time
87 FFFF06 00000002 87 PUL_TIMA4 .REW.W 1 ;Pulse set/reset time
88 FFFF08 00000002 88 PUL_TIMB4 .RES.W 1 ;Pulse set/reset time
89 ;

```

```

90 ;*****
91 ;*
92 ;*          SYMBOL DEFINITIONS
93 ;*
94 ;*****
95 ;
96      00FFFF60      96 TSTR      .EQU      H'FFFF60 ;Timer start register
97      00FFFF63      97 TFCR      .EQU      H'FFFF63 ;Timer function control
;register
98      00FFFF90      98 TOER      .EQU      H'FFFF90 ;Timer output master enable
;register
99      00FFFF91      99 TOCR      .EQU      H'FFFF91 ;Timer output control register
100 ;
101      00FFFF6E      101 TCR1      .EQU      H'FFFF6E ;Timer control register 0
102      00FFFF6F      102 TIOR1     .EQU      H'FFFF6F ;Timer I/O control register 0
103 ;
104      00FFFF82      104 TCR3      .EQU      H'FFFF82 ;Timer control register 3
105      00FFFF86      105 TCNT3     .EQU      H'FFFF86 ;Timer counter 3
106      00FFFF88      106 GRA3      .EQU      H'FFFF88 ;General register A3
107      00FFFF8A      107 GRB3      .EQU      H'FFFF8A ;General register B3
108 ;
109      00FFFF92      109 TCR4      .EQU      H'FFFF92 ;Timer control register 4
110      00FFFF96      110 TCNT4     .EQU      H'FFFF96 ;Timer counter 4
111      00FFFF98      111 GRA4      .EQU      H'FFFF98 ;General register A4
112      00FFFF9A      112 GRB4      .EQU      H'FFFF9A ;General register B4
113 ;
114 ;*****
115 ;*
116 ;*          MAIN PROGRAM : CMPWM3MN
117 ;*
118 ;*****
119 ;
120      010000      120          .SECTION  PROG, CODE, LOCATE=H'010000
121 ;
122      00010000      122 CMPWM3MN .EQU      $
123      010000      7A0700FFFEFE 123      MOV.L      #H'FFFEFE, SP ;Initialize stack pointer
124      010006      F8E0          124      MOV.B      #B'11100000, R0L
125      010008      3860          125      MOV.B      R0L, @TSTR      ;Stop TCNT3,4
126      01000A      F880          126      MOV.B      #B'10000000, R0L
127      01000C      3882          127      MOV.B      R0L, @TCR3     ;Initialize TCR3,4
128      01000E      3892          128      MOV.B      R0L, @TCR4
129 ;
130      010010      F88D          130      MOV.B      #B'10001101, R0L
131      010012      386F          131      MOV.B      R0L, @TIOR1   ;Initialize TIOR1
132      010014      F8E0          132      MOV.B      #B'11100000, R0L
133      010016      3863          133      MOV.B      R0L, @TFCR   ;Set complementary PWM mode

```

```

134 010018 79000000      134      MOV.W      #H'0000,R0
135 01001C 6B80FF96      135      MOV.W      R0,@TCNT4      ;Initialize TCNT4
136                                     136 ;
137 010020 6B00FF00      137      MOV.W      @NON_OVR,R0
138 010024 6B80FF86      138      MOV.W      R0,@TCNT3      ;Set non-overlap time
139                                     139 ;
140 010028 6B00FF02      140      MOV.W      @PUL_CYC,R0
141 01002C 6B80FF88      141      MOV.W      R0,@GRA3      ;Set pulse cycle time
142 010030 6B00FF06      142      MOV.W      @PUL_TIMA4,R0
143 010034 6B80FF98      143      MOV.W      R0,@GRA4      ;Set pulse set/reset time
144 010038 6B00FF04      144      MOV.W      @PUL_TIMB3,R0
145 01003C 6B80FF8A      145      MOV.W      R0,@GRB3      ;Set pulse set/reset time
146 010040 6B00FF08      146      MOV.W      @PUS_TIMB4,R0
147 010044 6B80FF9A      147      MOV.W      R0,@GRB4      ;Set pulse set/reset time
148                                     148 ;
149 01004B F8EF      149      MOV.B      #B'11101111,R0L
150 01004A 3891      150      MOV.B      R0L,@TOCR      ;Set outside trigger
                                     ;disable
151 01004C F8FF      151      MOV.B      #B'11111111,R0L
152 01004E 3890      152      MOV.B      R0L,@TOER      ;Set master enable all
                                     ;port
153                                     153 ;
154 010050 F8F8      154      MOV.B      #B'11111000,R0L
155 010052 3860      155      MOV.B      R0L,@TSTR      ;Set TCNT1,3,4 start data
156                                     156
157 010054 40FE      157 CMPWM3MN99 BRA      CMPWM3MN99
158                                     158
159                                     159      .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```

2.5 Long Cycle Pulse Output

MCU: H8/3003

Function: ITU (PWM mode)

2.5.1 Specifications

- Conducts 32-bit counter operation, and varies pulse high width to output long cycle pulses with variable duty (see figure 2.22)
- Duty can be set from 0% to 100% with a resolution of 1/65535
- During 16 MHz operation, the pulse cycle can be set in units of 4.09 ms, from 8.19 ms to 268.4 s

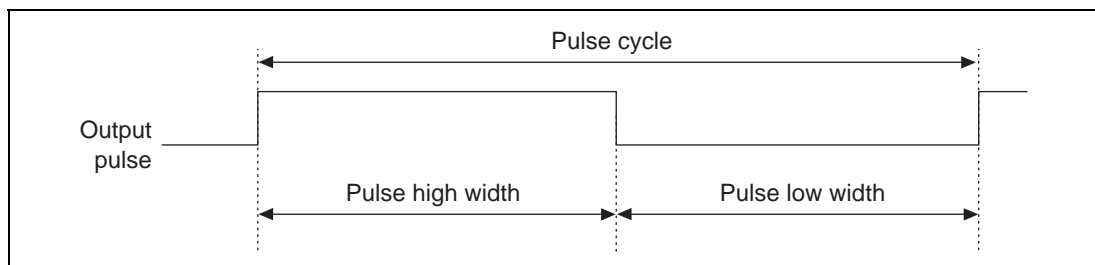


Figure 2.22 Long Cycle Pulse Output Example

2.5.2 Description of Functions Used

With this sample task, a 32-bit counter is operated using ch0, 3, and a long cycle pulse is output from ch3.

Figure 2.23 shows a block diagram of ITU/ch0 used in this sample task. ch0 uses the following functions to generate an external clock for input to ch3's timer counter.

- Automatic pulse output function in hardware without software intervention (output compare)
- Output reversal with each compare/match (toggle output)

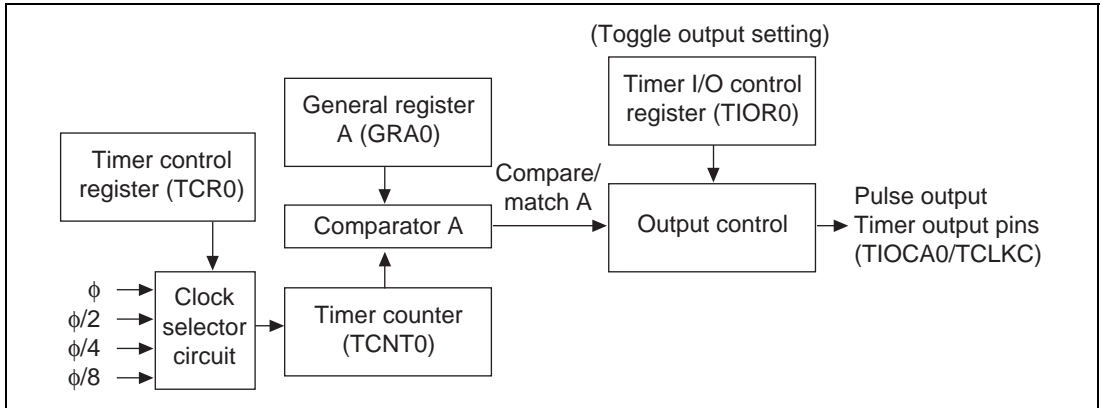


Figure 2.23 ITU/ch0 Block Diagram

Figure 2.24 shows a block diagram of ITU/ch3 used with this sample task. ch3 uses the following function to output duty pulses.

- Function that has the pulse output pin automatically become the PWM pin (PWM mode)

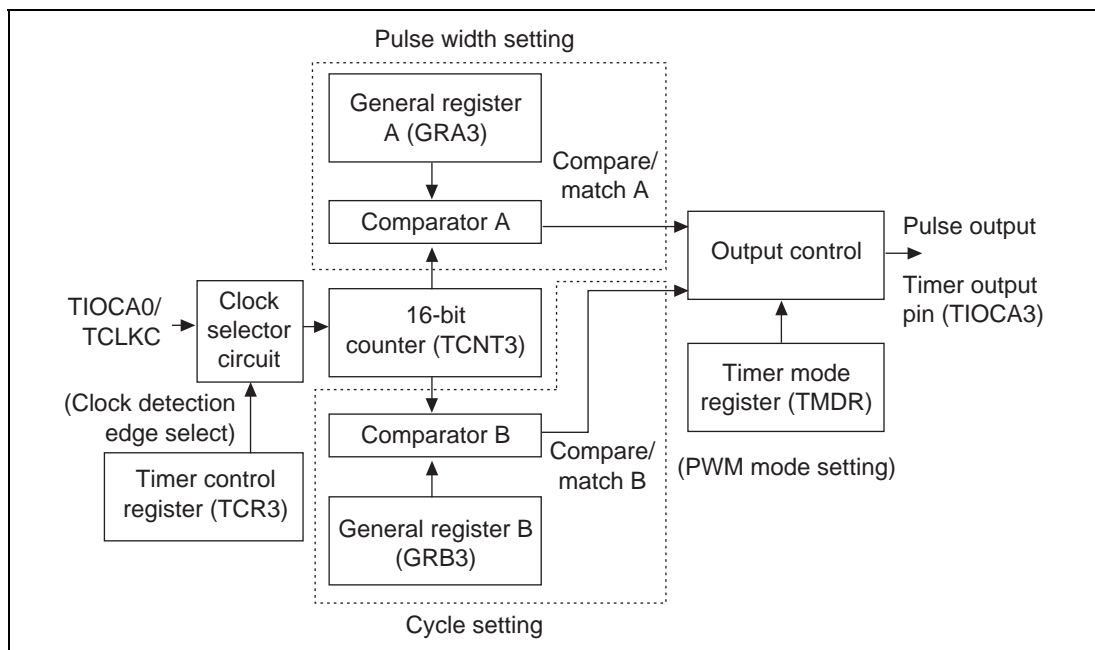


Figure 2.24 ITU/ch3 Block Diagram

Table 2.22 shows the function allocation for this sample task. ITU functions are allocated to generate the 32-bit timer counter.

Table 2.22 ITU Function Allocation

ITU Function	Function	
TMDR	Sets to PWM mode	
ch0	TCR0	Selects clock input to TCNT and counter clear factor
	TIOR0	Switches pulse output level
	TIOCA0	Outputs pulse
	GRA0	Sets pulse 1/2 cycle
ch3	TCR3	Selects clock input to TCNT and counter clear factor
	TIOCA3	Outputs pulse
	GRA3	Sets pulse width
	GRB3	Sets pulse cycle

2.5.3 Description of Operations

Figure 2.25 shows the ITU/ch0 principle of operation. Pulses are output and the ch3 external clock is generated by H8/3003 hardware and software processing.

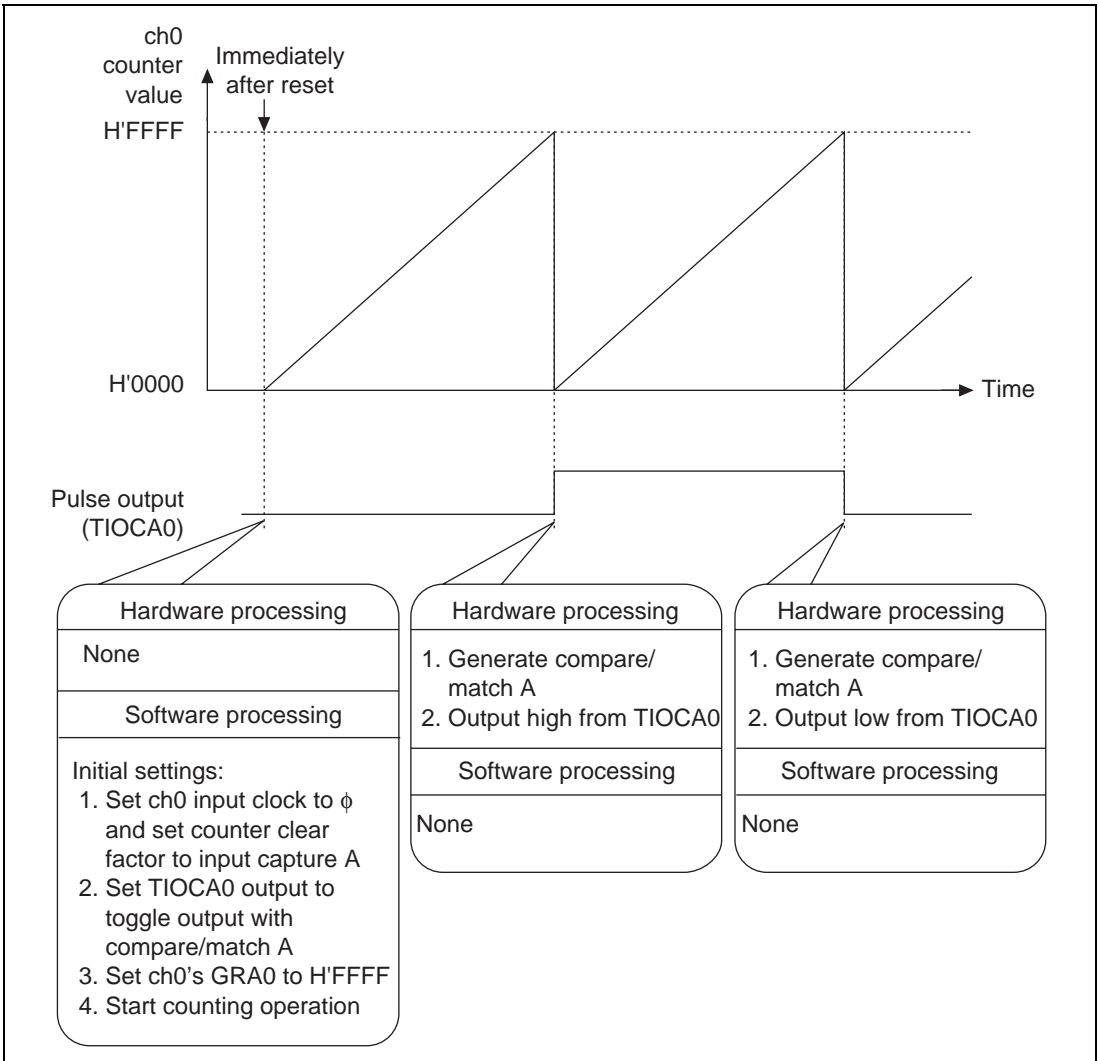


Figure 2.25 Principle of Pulse Output Operation (ITU/ch0)

Figure 2.26 shows the ITU/ch0/ch3 principle of operation. Long cycle pulses are output by H8/3003 hardware and software processing.

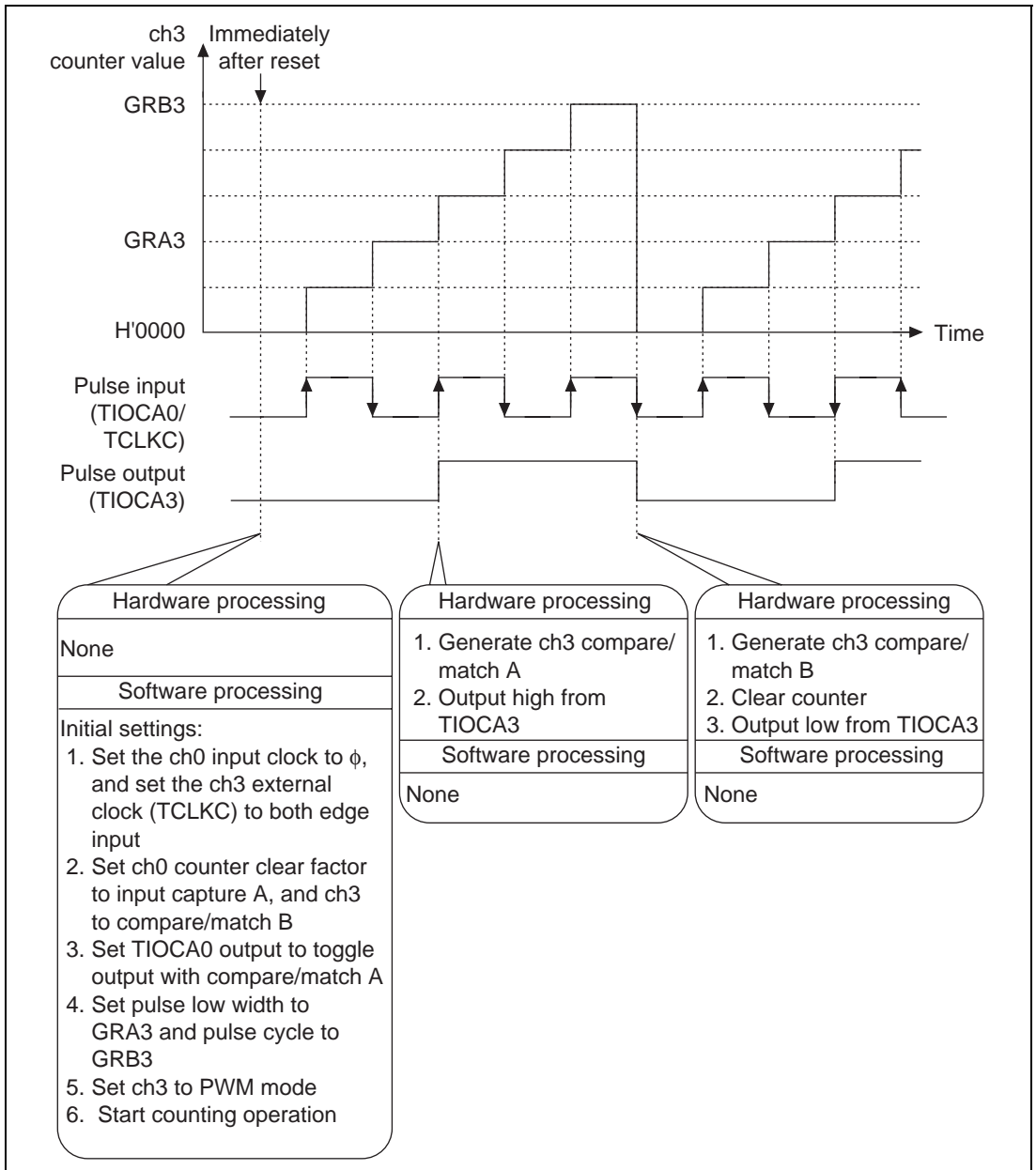


Figure 2.26 Principle of Pulse Output Operation (ITU/ch0, ch3)

2.5.4 Description of Software

The software for long cycle pulse output is described below.

Table 2.23 Description of Modules

Module Name	Label Name	Function
Main routine	LPULMN	32-bit counter operation using ch0/ch3 counters, and output of long cycle pulse.

Table 2.24 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
LPUL_WID	Setting of timer value equivalent to output pulse low width. Pulse low width is determined from the following formula: Pulse low width (ms) = timer value × external clock (4.09 ms during 16 MHz operation)	1 word	Main routine	Input
LPUL_CYC	Setting of timer value equivalent to the output pulse cycle. Cycle is determined by the following formula: Cycle (ms) = timer value × external clock (4.09 ms during 16 MHz operation)	1 word	Main routine	Input

Table 2.25 Description of Internal Registers

Register Name	Function	Module Name	
TSTR	Sets timer counter operation/prohibit	Main routine	
TMDR	Sets ch3 to PWM mode	Main routine	
ch0	TCR0	Selects input clock to TCNT and sets timer counter clear factor	Main routine
	TIOR0	Sets output level from TIOCA0 during compare match A	Main routine
	GRA0	Sets 1/2 cycle of pulse output to ch3	Main routine
ch3	TCR3	Selects input clock to TCNT and sets timer counter clear factor	Main routine
	GRA3	Sets pulse low width	Main routine
	GRB3	Sets pulse cycle	Main routine

Table 2.26 Description of General Registers

Module Name	Register Name	Function
Main routine	R0	Used as work register during data setting.

Description of RAM: RAM not used other than for arguments in this sample task.

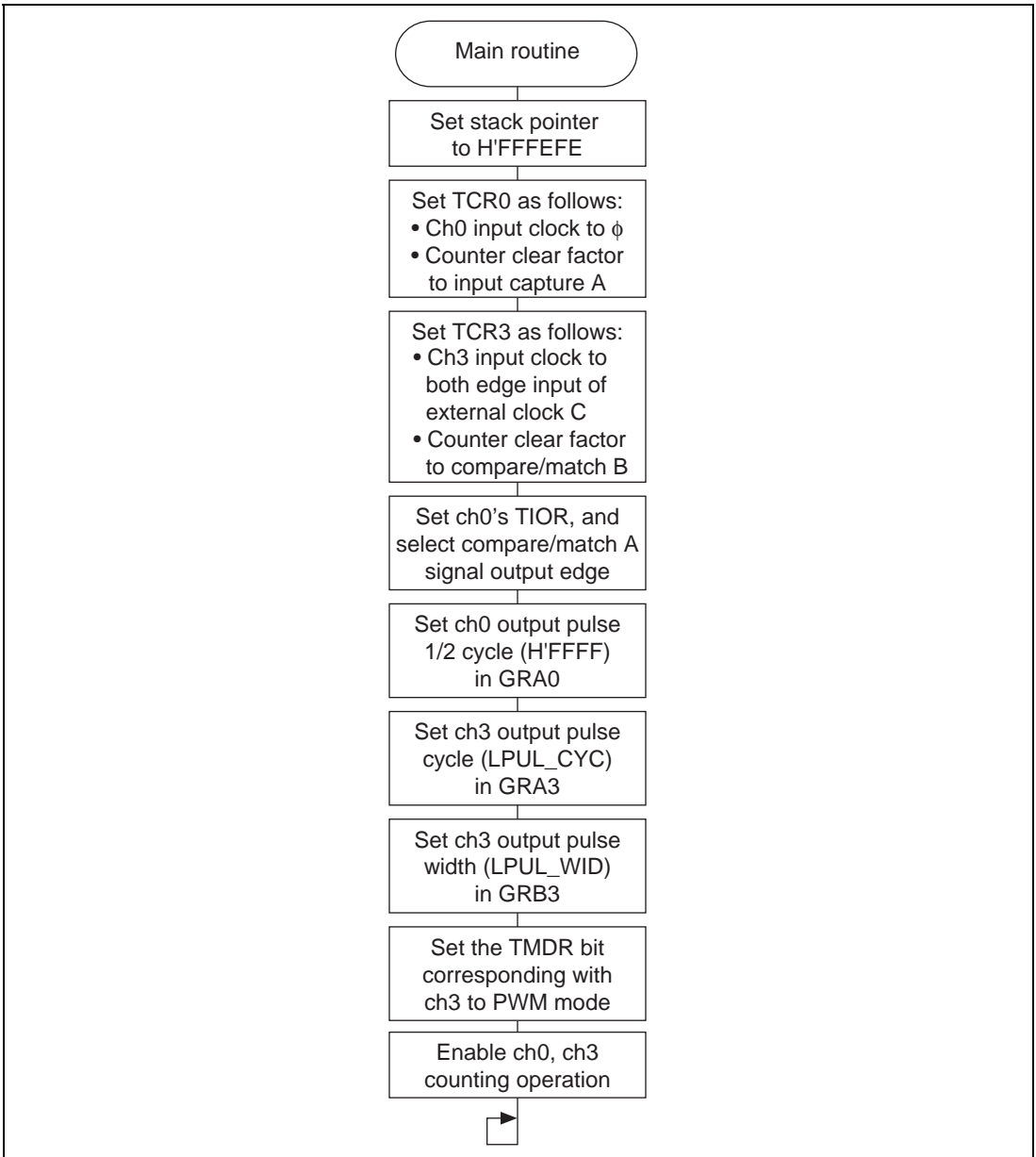


Figure 2.27 Main Routine Flowchart

2.5.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION  VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES      .DATA.L   LPULMN
10         10 ;
11 00001C   11         .ORG      H'00001C
12 00001C 00010000  12 NMI     .DATA.L   LPULMN
13 000020 00010000  13 TRAPA0  .DATA.L   LPULMN
14 000024 00010000  14 TRAPA1  .DATA.L   LPULMN
15 000028 00010000  15 TRAPA2  .DATA.L   LPULMN
16 00002C 00010000  16 TRAPA3  .DATA.L   LPULMN
17         17 ;
18 000030   18         .ORG      H'000030
19 000030 00010000  19 IRQ0    .DATA.L   LPULMN
20 000034 00010000  20 IRQ1    .DATA.L   LPULMN
21 000038 00010000  21 IRQ2    .DATA.L   LPULMN
22 00003C 00010000  22 IRQ3    .DATA.L   LPULMN
23 000040 00010000  23 IRQ4    .DATA.L   LPULMN
24 000044 00010000  24 IRQ5    .DATA.L   LPULMN
25 000048 00010000  25 IRQ6    .DATA.L   LPULMN
26 00004C 00010000  26 IRQ7    .DATA.L   LPULMN
27 000050 00010000  27 WOVI0   .DATA.L   LPULMN
28 000054 00010000  28 CMI     .DATA.L   LPULMN
29         29 ;
30 000060   30         .ORG      H'000060
31 000060 00010000  31 IMIA0   .DATA.L   LPULMN
32 000064 00010000  32 IMIB0   .DATA.L   LPULMN
33 000068 00010000  33 OVI0    .DATA.L   LPULMN
34         34 ;
35 000070   35         .ORG      H'000070
36 000070 00010000  36 IMIA1   .DATA.L   LPULMN
37 000074 00010000  37 IMIB1   .DATA.L   LPULMN
38 000078 00010000  38 OVI1    .DATA.L   LPULMN
39         39 ;
40 000080   40         .ORG      H'000080
41 000080 00010000  41 IMIA2   .DATA.L   LPULMN
42 000084 00010000  42 IMIB2   .DATA.L   LPULMN
43 000088 00010000  43 OVI2    .DATA.L   LPULMN

```

```

44                                     44 ;
45 000090                             45          .ORG          H'000090
46 000090 00010000                     46 IMIA3      .DATA.L      LPULMN
47 000094 00010000                     47 IMIB3      .DATA.L      LPULMN
48 000098 00010000                     48 OVI3       .DATA.L      LPULMN
49                                     49 ;
50 0000A0                             50          .ORG          H'0000A0
51 0000A0 00010000                     51 IMIA4      .DATA.L      LPULMN
52 0000A4 00010000                     52 IMIB4      .DATA.L      LPULMN
53 0000A8 00010000                     53 OVI4       .DATA.L      LPULMN
54                                     54 ;
55 0000B0                             55          .ORG          H'0000B0
56 0000B0 00010000                     56 DEND0A     .DATA.L      LPULMN
57 0000B4 00010000                     57 DEND0B     .DATA.L      LPULMN
58 0000B8 00010000                     58 DEND1A     .DATA.L      LPULMN
59 0000BC 00010000                     59 DEND1B     .DATA.L      LPULMN
60 0000C0 00010000                     60 DEND2A     .DATA.L      LPULMN
61 0000C4 00010000                     61 DEND2B     .DATA.L      LPULMN
62 0000C8 00010000                     62 DEND3A     .DATA.L      LPULMN
63 0000CC 00010000                     63 DEND3B     .DATA.L      LPULMN
64 0000D0 00010000                     64 ERI0       .DATA.L      LPULMN
65 0000D4 00010000                     65 RXI0       .DATA.L      LPULMN
66 0000D8 00010000                     66 TXI0       .DATA.L      LPULMN
67 0000DC 00010000                     67 TEI0       .DATA.L      LPULMN
68 0000E0 00010000                     68 ERI1       .DATA.L      LPULMN
69 0000E4 00010000                     69 RXI1       .DATA.L      LPULMN
70 0000E8 00010000                     70 TXI1       .DATA.L      LPULMN
71 0000EC 00010000                     71 TEI1       .DATA.L      LPULMN
72 0000F0 00010000                     72 ADI        .DATA.L      LPULMN
73                                     73 ;
74                                     74 ;*****
75                                     75 ;*                                     *
76                                     76 ;*          RAM ALLOCATION          *
77                                     77 ;*                                     *
78                                     78 ;*****
79                                     79 ;
80 FFFF00                             80          .SECTION    RAM,DATA,LOCATE=H'FFFF00
81                                     81 ;
82 FFFF00 00000002                     82 LPUL_CYC   .RES.W      1          ;Output pulse cycle time
83 FFFF02 00000002                     83 LPUL_WID   .RES.W      1          ;Output pulse low period time
84                                     84 ;
85                                     85 ;*****
86                                     86 ;*                                     *
87                                     87 ;*          SYMBOL DEFINTIONS      *
88                                     88 ;*                                     *
89                                     89 ;*****

```

```

90          90 ;
91          00FFFF60 91 TSTR .EQU H'FFFF60 ;Timer start register
92          00FFFF62 92 TMDR .EQU H'FFFF62 ;Timer mode register
93          93 ;
94          00FFFF64 94 TCR0 .EQU H'FFFF64 ;Timer control register 0
95          00FFFF65 95 TIOR0 .EQU H'FFFF65 ;Timer counter 0
96          00FFFF6A 96 GRA0 .EQU H'FFFF6A ;General register A0
97          97 ;
98          00FFFF82 98 TCR3 .EQU H'FFFF82 ;Timer control register 1
99          00FFFF88 99 GRA3 .EQU H'FFFF88 ;General register A1
100         00FFFF8A 100 GRB3 .EQU H'FFFF8A ;General register B1
101         101 ;
102         102 ;*****
103         103 ;* *
104         104 ;* MAIN PROGRAM : LPULMN *
105         105 ;* *
106         106 ;*****
107         107 ;
108 010000 108 .SECTION PROG, CODE, LOCATE=H'010000
109         109 ;
110         00010000 110 LPULMN: .EQU $
111 010000 7A0700FFFEFE 111 MOV.L #H'FFFEFE, SP ;Initialize stack pointer
112 010006 F880 112 MOV.B #B'10000000, R0L
113 010008 3864 113 MOV.B R0L, @TCR0 ;Initialize TCR0
114 01000A F8D6 114 MOV.B #B'11010110, R0L
115 01000C 3882 115 MOV.B R0L, @TCR3 ;Initialize TCR3
116         116 ;
117 01000E F883 117 MOV.B #B'10000011, R0L
118 010010 3865 118 MOV.B R0L, @TIOR0 ;Initialize TIOR0
119         119 ;
120 010012 7900FFFF 120 MOV.W #H'FFFF, R0
121 010016 6B80FF6A 121 MOV.W R0, @GRA0 ;Set outside signal cycle
;time
122 01001A 6B00FF00 122 MOV.W @LPUL_CYC, R0
123 01001E 6B80FF8A 123 MOV.W R0, @GRB3 ;Set pulse cycle time
124 010022 6B00FF02 124 MOV.W @LPUL_WID, R0
125 010026 6B80FF88 125 MOV.W R0, @GRA3 ;Set pulse low period
;time
126         126 ;
127 01002A F888 127 MOV.B #B'10001000, R0L
128 01002C 3862 128 MOV.B R0L, @TMDR ;Set PWM mode (ch1)
129 01002E F8E9 129 MOV.B #B'11101001, R0L
130 010030 3860 130 MOV.B R0L, @TSTR ;Start TCNT0, 3
131         131
132 010032 40FE 132 LPULMN99 BRA LPULMN99
133         133

```


134

134

.END

*****TOTAL ERRORS 0

*****TOTAL WARNINGS 0

2.6 Positive/Negative PWM Three-Phase Output

MCU: H8/3003

Function: ITU (reset sync PWM mode)

2.6.1 Specifications

As shown in figure 2.28, this sample task varies pulse high width, and outputs a pulse of variable duty (duty pulse) in a 3-phase output with positive and negative phases.

During 16 MHz operation, the output pulse cycle can be optionally set from 125 ns to 4.09 ms.

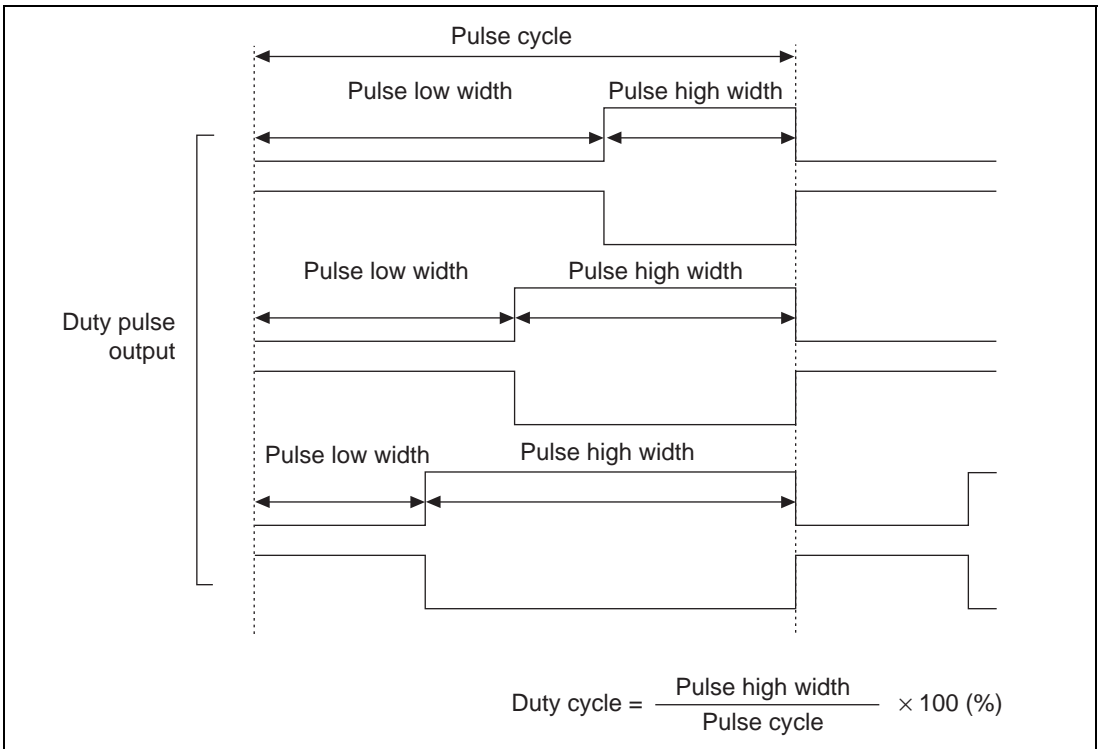


Figure 2.28 Positive/Negative PWM 3-Phase Output Waveforms

2.6.2 Description of Functions Used

This sample task uses ITU ch3/ch4 to give 3-phase output of positive and negative phase PWM waveforms.

Figure 2.29 shows a block diagram of the ITU used in this sample task.

The following ITU functions are used in this task:

- Timer counter clear during compare/match
- Output of positive and negative phase PWM waveforms with common relation of waveform variation points due to a combination of ch3 and ch4 (reset sync PWM mode)

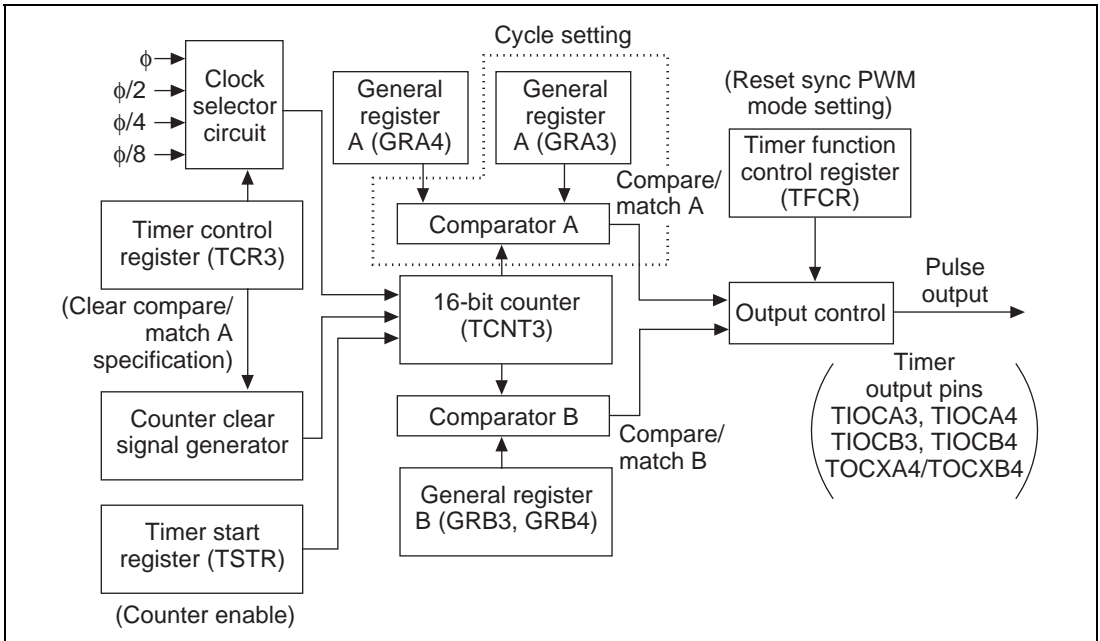


Figure 2.29 ITU Block Diagram

Table 2.27 shows the ITU function allocation for positive/negative phase PWM waveform 3-phase output with this sample task.

Table 2.27 ITU Function Allocation

ITU Function		Function
TSTR		Enables/disables TCNT count operation
TFCR		Sets to reset sync PWM mode
ch3	TCR3	Sets counter clock and counter clear factor
	TIOCA3	Outputs positive phase pulse
	TIOCAB3	Outputs TIOCA negative pulse
	GRA3	Sets output pulse cycle
ch4	TIOCA4	Outputs positive phase pulse
	TOCXA4	
	TIOCB4	Outputs TIOCB and TOCXB negative phase pulses
	TOCXB4	
	GRA4	Sets variation timing for pulses output from TIOCA and TOCXA pins
GRB4		Sets variation timing for pulses output from TIOCB and TOCXB pins

2.6.3 Description of Operations

Figure 2.30 shows the operation principle. Positive and negative phase PWM waveforms are output in 3-phases due to H8/3003 hardware and software processing.

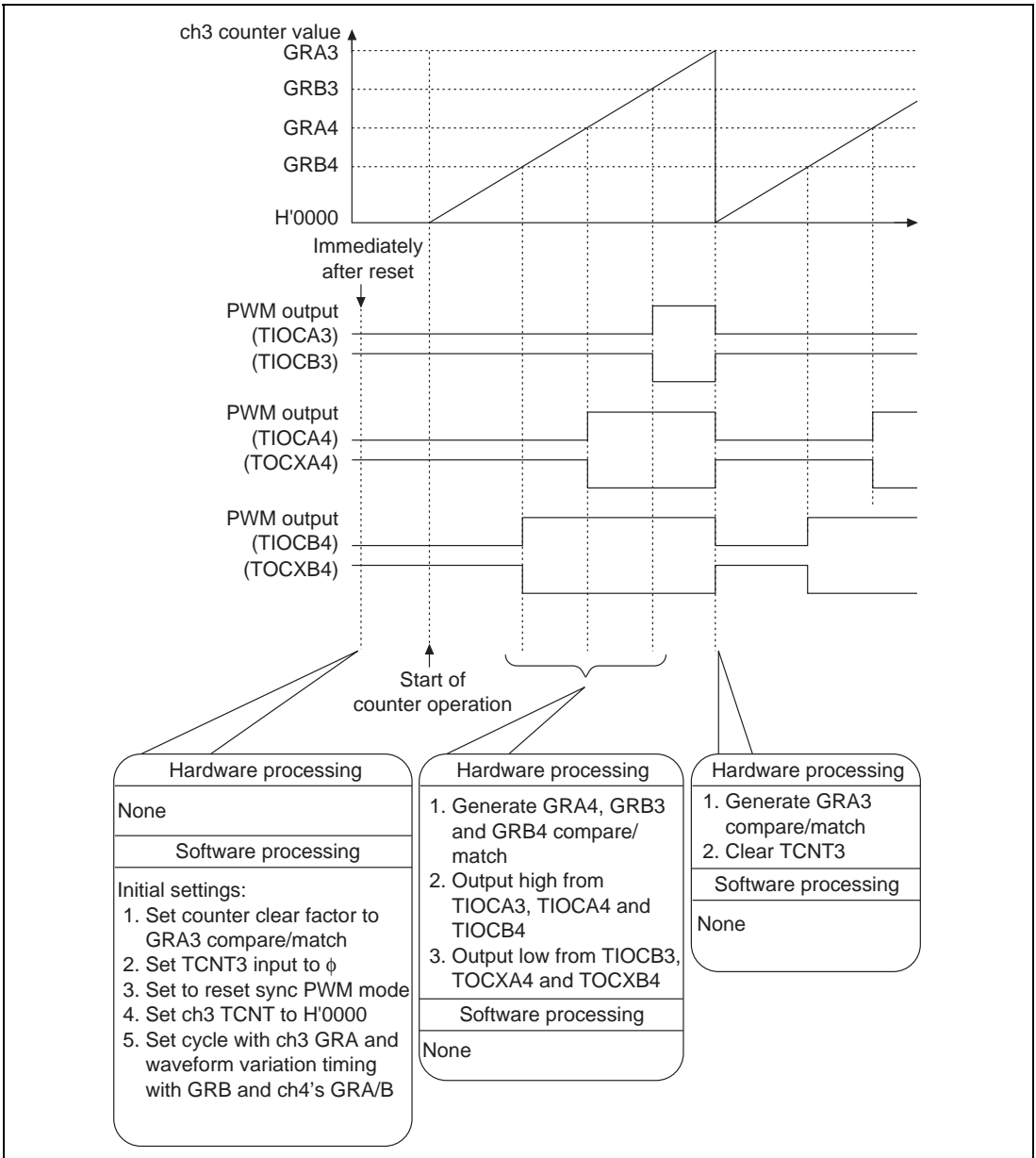


Figure 2.30 Operation Principle of 3-Phase PWM Output

2.6.4 Description of Software

The software for positive/negative PWM three-phase output is described below.

Table 2.28 Description of Modules

Module Name	Label Name	Function
Main routine	NRPWM3MN	Sets output pulse waveform variation timing in GRA4, GRB3 and GRB4, sets the cycle in GRA3, and provides 3-phase output of positive/negative phase PWM waveforms.

Table 2.29 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
PUL_CYC	Sets timer value equivalent to PWM pulse cycle. Cycle is determined from the following formula: Cycle (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)	1 word	Main routine	Input
PUL_TIMB3 PUL_TIMA4 PUL_TIMB4	Outputs timer value equivalent to PWM pulse waveform output timing. Waveform output timing is determined from the following formula: Waveform output timing (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)	1 word	Main routine	Input

Table 2.30 Description of Internal Registers

Register Name	Function	Module Name	
TSTR	Sets timer counter operation/prohibit	Main routine	
TFCR	Sets ch3/ch4 to reset sync PWM mode	Main routine	
ch3	TCR3	Selects counter clock and sets counter clear factor to GRA compare/match	Main routine
	TCNT	Sets initial value H'0000	Main routine
	GRA3	Sets pulse cycle	Main routine
	GRB3	Sets PWM output waveform variation timing	Main routine
ch4	GRA4	Sets PWM output waveform variation timing	Main routine
	GRB4	Sets PWM output waveform variation timing	Main routine

Table 2.31 Description of General Registers

Module Name	Register Name	Function
Main routine	R0	Used as work register during data setting.

Description of RAM: RAM not used other than for arguments in this sample task.

2.6.5 Flowchart

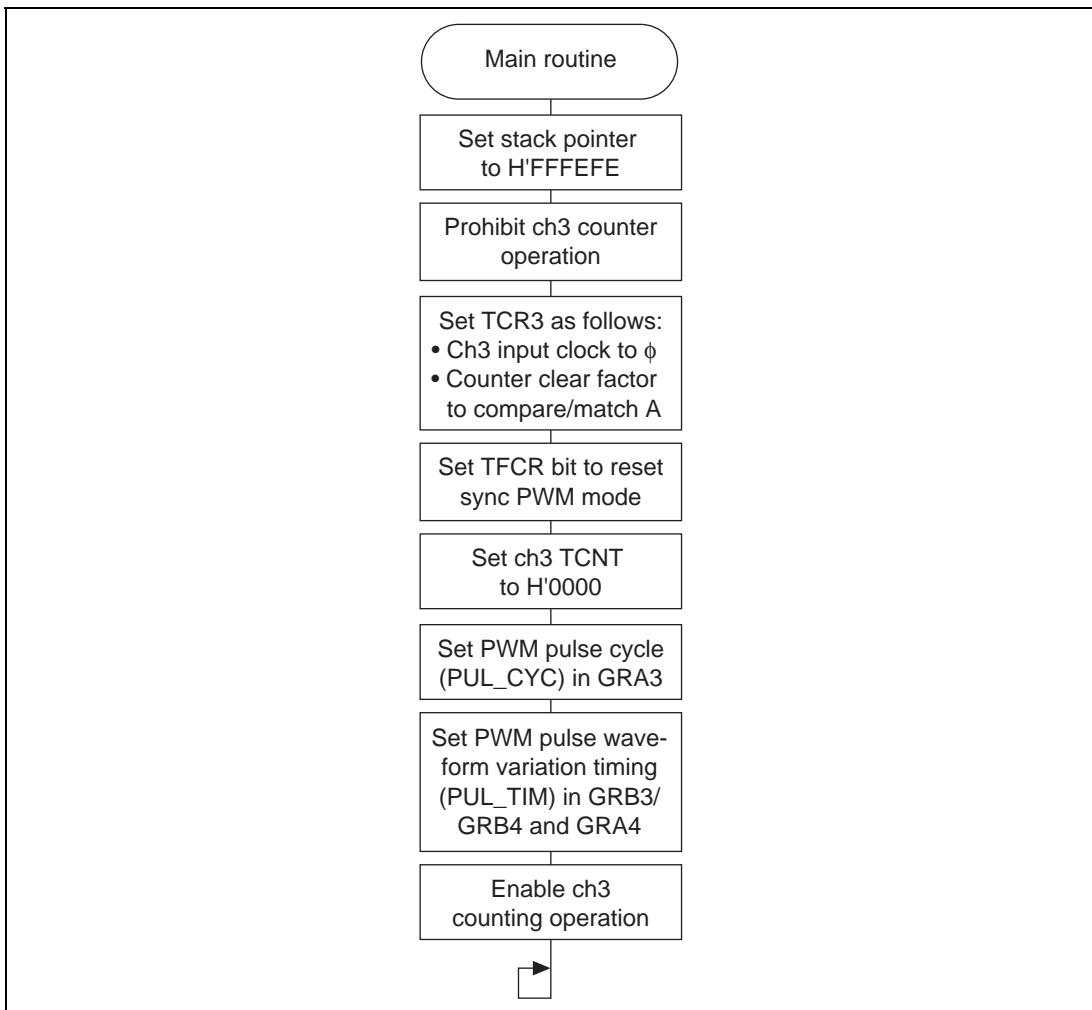


Figure 2.31 Main Routine Flowchart

2.6.6 Program List

```

1      1 ;*****
2      2 ;*
3      3 ;*      VECTOR ADDRESS
4      4 ;*
5      5 ;*****
6      6 ;
7      7      .SECTION  VECT, CODE, LOCATE=H'000000
8      8 ;
9      9 RES      .DATA.L  NRPWM3MN
10     10 ;
11     11      .ORG      H'00001C
12     12 NMI      .DATA.L  NRPWM3MN
13     13 TRAPA0   .DATA.L  NRPWM3MN
14     14 TRAPA1   .DATA.L  NRPWM3MN
15     15 TRAPA2   .DATA.L  NRPWM3MN
16     16 TRAPA3   .DATA.L  NRPWM3MN
17     17 ;
18     18      .ORG      H'000030
19     19 IRQ0     .DATA.L  NRPWM3MN
20     20 IRQ1     .DATA.L  NRPWM3MN
21     21 IRQ2     .DATA.L  NRPWM3MN
22     22 IRQ3     .DATA.L  NRPWM3MN
23     23 IRQ4     .DATA.L  NRPWM3MN
24     24 IRQ5     .DATA.L  NRPWM3MN
25     25 IRQ6     .DATA.L  NRPWM3MN
26     26 IRQ7     .DATA.L  NRPWM3MN
27     27 WOVI0    .DATA.L  NRPWM3MN
28     28 CMI      .DATA.L  NRPWM3MN
29     29 ;
30     30      .ORG      H'000060
31     31 IMIA0    .DATA.L  NRPWM3MN
32     32 IMIB0    .DATA.L  NRPWM3MN
33     33 OVI0     .DATA.L  NRPWM3MN
34     34 ;
35     35      .ORG      H'000070
36     36 IMIA1    .DATA.L  NRPWM3MN
37     37 IMIB1    .DATA.L  NRPWM3MN
38     38 OVI1     .DATA.L  NRPWM3MN
39     39 ;
40     40      .ORG      H'000080
41     41 IMIA2    .DATA.L  NRPWM3MN
42     42 IMIB2    .DATA.L  NRPWM3MN
43     43 OVI2     .DATA.L  NRPWM3MN

```



```

44 ;
45 000090 00010000 45 .ORG H'000090
46 000090 00010000 46 IMIA3 .DATA.L NRPWM3MN
47 000094 00010000 47 IMIB3 .DATA.L NRPWM3MN
48 000098 00010000 48 OVI3 .DATA.L NRPWM3MN
49 ;
50 0000A0 00010000 50 .ORG H'0000A0
51 0000A0 00010000 51 IMIA4 .DATA.L NRPWM3MN
52 0000A4 00010000 52 IMIB4 .DATA.L NRPWM3MN
53 0000A8 00010000 53 OVI4 .DATA.L NRPWM3MN
54 ;
55 0000B0 00010000 55 .ORG H'0000B0
56 0000B0 00010000 56 DEND0A .DATA.L NRPWM3MN
57 0000B4 00010000 57 DEND0B .DATA.L NRPWM3MN
58 0000B8 00010000 58 DEND1A .DATA.L NRPWM3MN
59 0000BC 00010000 59 DEND1B .DATA.L NRPWM3MN
60 0000C0 00010000 60 DEND2A .DATA.L NRPWM3MN
61 0000C4 00010000 61 DEND2B .DATA.L NRPWM3MN
62 0000C8 00010000 62 DEND3A .DATA.L NRPWM3MN
63 0000CC 00010000 63 DEND3B .DATA.L NRPWM3MN
64 0000D0 00010000 64 ERI0 .DATA.L NRPWM3MN
65 0000D4 00010000 65 RXI0 .DATA.L NRPWM3MN
66 0000D8 00010000 66 TXI0 .DATA.L NRPWM3MN
67 0000DC 00010000 67 TEI0 .DATA.L NRPWM3MN
68 0000E0 00010000 68 ERI1 .DATA.L NRPWM3MN
69 0000E4 00010000 69 RXI1 .DATA.L NRPWM3MN
70 0000E8 00010000 70 TXI1 .DATA.L NRPWM3MN
71 0000EC 00010000 71 TEI1 .DATA.L NRPWM3MN
72 0000F0 00010000 72 ADI .DATA.L NRPWM3MN
73 ;
74 ;*****
75 ;* *
76 ;* RAM ALLOCATION *
77 ;* *
78 ;*****
79 ;
80 FFFF00 00000002 80 .SECTION RAM,DATA,LOCATE=H'FFFF00
81 ;
82 FFFF00 00000002 82 PUL_CYC .RES.W 1 ;Pulse cycle Time
83 ;
84 FFFF02 00000002 84 PUL_TIMA4 .RES.W 1 ;Pulse change timing GRB3
85 FFFF04 00000002 85 PUL_TIMB3 .RES.W 1 ;Pulse change timing GRA4
86 FFFF06 00000002 86 PUL_TIMB4 .RES.W 1 ;Pulse change timing GRB4
87 ;

```

```

88 ;*****
89 ;*
90 ;*      SYMBOL DEFINITIONS
91 ;*
92 ;*****
93 ;
94      00FFFF60      94 TSTR      .EQU      H'FFFF60      ;Timer start register
95      00FFFF63      95 TFCR      .EQU      H'FFFF63      ;Timer function control
                                           ;register
96 ;
97      00FFFF82      97 TCR3      .EQU      H'FFFF82      ;Timer control register3
98      00FFFF86      98 TCNT3     .EQU      H'FFFF86      ;Timer counter3
99      00FFFF88      99 GRA3      .EQU      H'FFFF88      ;General register A3
100     00FFFF8A      100 GRB3     .EQU      H'FFFF8A      ;General register B3
101 ;
102     00FFFF98      102 GRA4     .EQU      H'FFFF98      ;General register A4
103     00FFFF9A      103 GRB4     .EQU      H'FFFF9A      ;General register B4
104 ;
105 ;*****
106 ;*
107 ;*      MAIN PROGRAM : NRPWM3MN
108 ;*
109 ;*****
110 ;
111 010000      111      .SECTION  PROG,CODE,LOCATE=H'010000
112 ;
113      00010000      113 NRPWM3MN .EQU $
114 010000      7A0700FFFEFE      114      MOV.L      #H'FFFEFE,SP ;Initialize stack pointer
115 010006      F8E0      115      MOV.B      #B'11100000,R0L
116 010008      3860      116      MOV.B      R0L,@TSTR ;Set TCNT3 stop data
117 01000A      F8A0      117      MOV.B      #B'10100000,R0L
118 01000C      3882      118      MOV.B      R0L,@TCR3 ;Initialize TCR3
119 ;
120 01000E      F8F0      120      MOV.B      #B'11110000,R0L
121 010010      3863      121      MOV.B      R0L,@TFCR ;Set reset-synchronized
mode
122 010012      79000000      122      MOV.W      #H'0000,R0
123 010016      6B80FF86      123      MOV.W      R0,@TCNT3 ;Initialize TCNT3
124 ;
125 01001A      6B00FF00      125      MOV.W      @PUL_CYC,R0
126 01001E      6B80FF88      126      MOV.W      R0,@GRA3 ;Set pulse cycle time
127 010022      6B00FF02      127      MOV.W      @PUL_TIMA4,R0
128 010026      6B80FF98      128      MOV.W      R0,@GRA4 ;Set pulse change timing
129 01002A      6B00FF04      129      MOV.W      @PUL_TIMB3,R0
130 01002E      6B80FF8A      130      MOV.W      R0,@GRB3 ;Set pulse change timing
131 010032      6B00FF06      131      MOV.W      @PUL_TIMB4,R0

```

```
132 010036 6B80FF9A      132      MOV.W      R0,@GRB4      ;Set pulse change timing
133
134 01003A F8E8          134      MOV.B      #B'11101000,R0L
135 01003C 3860          135      MOV.B      R0L,@TSTR      ;Start TCNT3
136
137 01003E 40FE          137 NRPWM3MN99  BRA      NRPWM3MN99
138
139
139      .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0
```

2.7 PWM Five-Phase Output

MCU: H8/3003

Function: ITU (sync/PWM mode)

2.7.1 Specifications

As shown in figure 2.32, pulse high width is varied to give a 5-phase output variable duty pulse.

Duty can be optionally set from ch0 to ch4.

$$\text{Duty} = \text{Pulse high width} / \text{Pulse cycle} \times 100 (\%)$$

During 16 MHz operation, the output pulse cycle can be set from 125 ns to 4.09 ms.

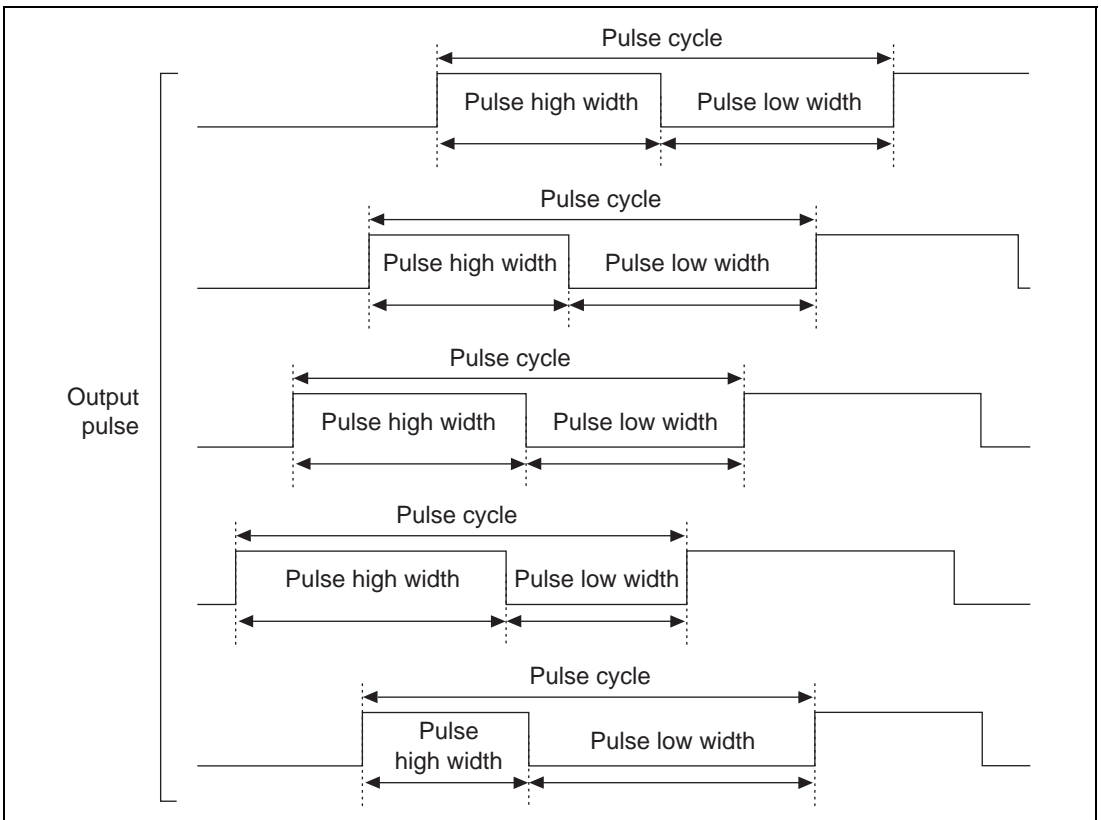


Figure 2.32 PWM 5-Phase Output Example

2.7.2 Description of Functions Used

In this sample task, using ITU ch0–ch4, timer counters are set to sync operation and 5-phase PWM waveforms are output.

Figure 2.33 shows the ITU block diagram for this sample task. The following ITU functions are used:

- Simultaneous clearing of multiple timer counters during compare/match (sync clear)
- Automatic switching of pulse output pin to PWM pin (PWM mode)

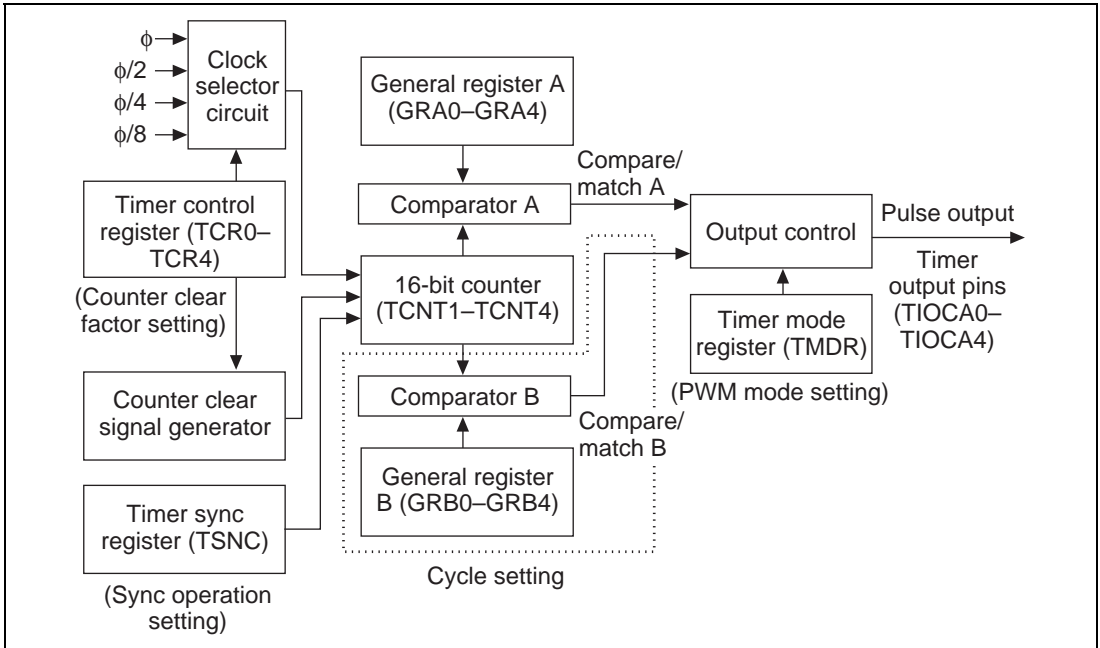


Figure 2.33 Sync Pulse Output Block Diagram

Table 2.32 shows the ITU functions allocated for 5-phase PWM waveform output.

Table 2.32 ITU Function Allocation

ITU Function		Function
TSNC		Sets synchronous operation from ch0–ch4
TMDR		Sets to PWM mode from ch0–ch4
ch0–ch4	TCR0–TCR4	Sets counter clock and counter clear factor
	TIOCA0–TIOCA4	Outputs pulse
	GRA0–GRA4	Sets pulse set timing
	GRB0–GRB4	Sets pulse cycle

2.7.3 Description of Operations

Figure 2.34 shows the ch0 principle of operation. Duty pulses are output due to H8/3003 hardware and software processing.

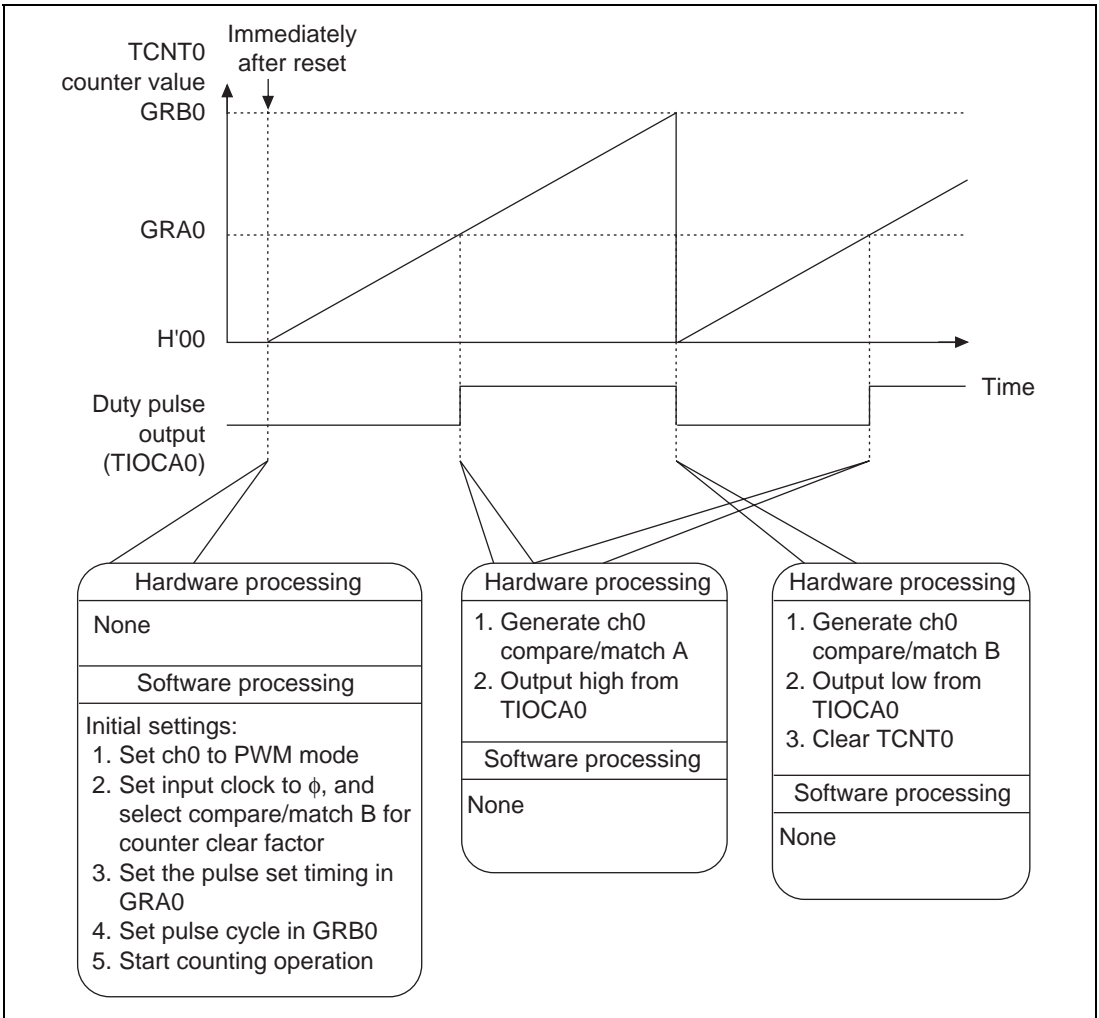


Figure 2.34 Principle of PWM Output Operation

Figure 2.35 shows the operation principle for ch0–ch4. A duty pulse synchronized with ch0 is output due to H8/3003 hardware and software processing.

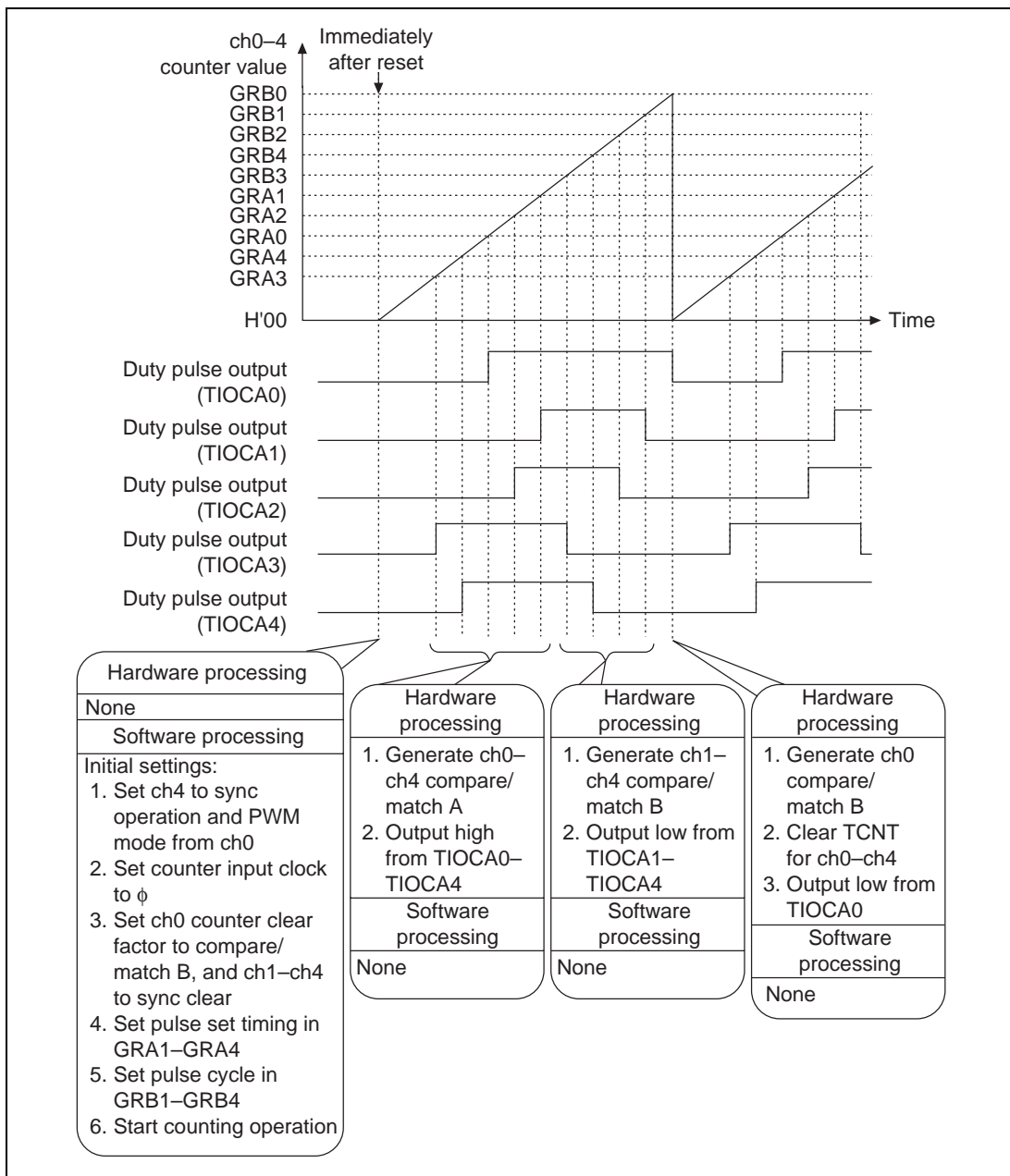


Figure 2.35 Principle of PWM 5-Phase Output Operation

2.7.4 Description of Software

The software for PWM five-phase output is described below.

Table 2.33 Description of Modules

Module Name	Label Name	Function
Main routine	PWM5MN	Mode setting, output pulse cycle and pulse width setting, RAM initial setting and 5-phase PWM pulse output.

Table 2.34 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
PUL_CYC 0–4	<p>Sets timer value equivalent to PWM pulse cycle.</p> <p>Cycle is determined from the following formula:</p> <p>Cycle (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)</p>	1 word	Main routine	Input
PUL_SET 0–4	<p>Sets timer value equivalent to PWM pulse set timing.</p> <p>Set timing is determined from the following formula.</p> <p>Set timing (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)</p>	1 word	Main routine	Input

Table 2.35 Description of Internal Registers

Register Name	Function	Module Name	
TSTR	Timer counter enable/disable setting	Main routine	
TSNC	Sets ch0–ch4 sync operation	Main routine	
TMDR	Sets ch0–ch4 to PWM mode	Main routine	
ch0	TCR0	Sets counter clear factor to compare/match B	Main routine
	GRA0	Sets pulse set timing	Main routine
	GRB0	Sets pulse cycle	Main routine
ch1–ch4	TCR1–TCR4	Sets counter clear factor to sync clear	Main routine
	GRA1–GRA4	Sets pulse set timing	Main routine
	GRB1–GRB4	Sets pulse cycle	Main routine

Table 2.36 Description of General Registers

Module Name	Register Name	Function
Main routine	R0	Used as work register during data setting

Description of RAM: RAM not used other than for arguments in this sample task.

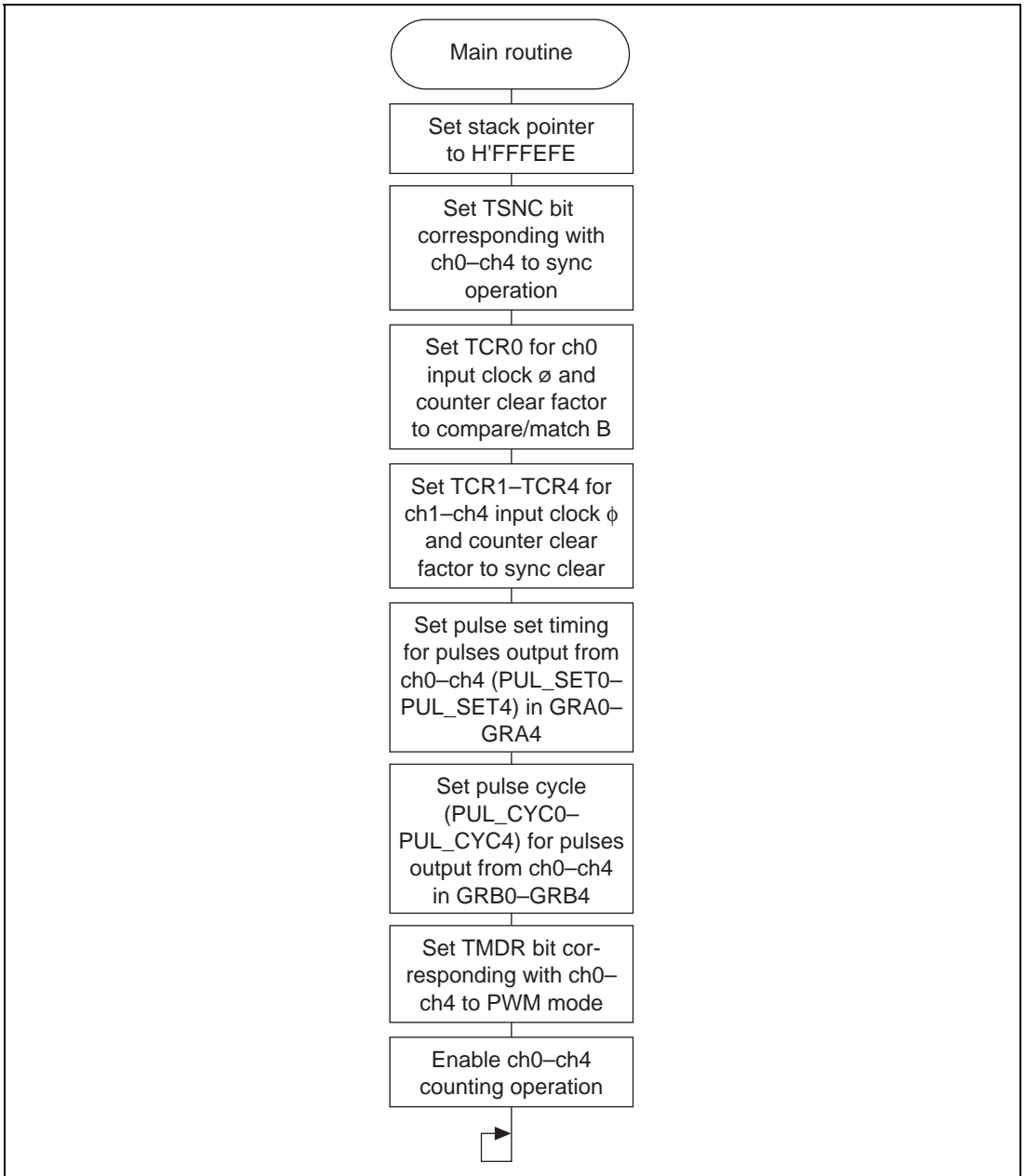


Figure 2.36 PWM Five-Phase Output Flowchart

2.7.6 Program List

```
1 ;*****
2 ;*
3 ;* VECTOR ADDRESS
4 ;*
5 ;*****
6 ;
7 .CPU 300HA
8 000000 8 ; .SECTION VECT, CODE, LOCATE=H'000000
9 ;
10 000000 00010000 10 RES .DATA.L PWM5MN
11 ;
12 00001C 12 .ORG H'00001C
13 00001C 00010000 13 NMI .DATA.L PWM5MN
14 000020 00010000 14 TRAPA0 .DATA.L PWM5MN
15 000024 00010000 15 TRAPA1 .DATA.L PWM5MN
16 000028 00010000 16 TRAPA2 .DATA.L PWM5MN
17 00002C 00010000 17 TRAPA3 .DATA.L PWM5MN
18 ;
19 00030 19 .ORG H'000030
20 000030 00010000 20 IRQ0 .DATA.L PWM5MN
21 000034 00010000 21 IRQ1 .DATA.L PWM5MN
22 000038 00010000 22 IRQ2 .DATA.L PWM5MN
23 00003C 00010000 23 IRQ3 .DATA.L PWM5MN
24 000040 00010000 24 IRQ4 .DATA.L PWM5MN
25 000044 00010000 25 IRQ5 .DATA.L PWM5MN
26 000048 00010000 26 IRQ6 .DATA.L PWM5MN
27 00004C 00010000 27 IRQ7 .DATA.L PWM5MN
28 000050 00010000 28 WOVI0 .DATA.L PWM5MN
29 000054 00010000 29 CMI .DATA.L PWM5MN
30 ;
31 000060 31 .ORG H'000060
32 000060 00010000 32 IMIA0 .DATA.L PWM5MN
33 000064 00010000 33 IMIB0 .DATA.L PWM5MN
34 000068 00010000 34 OVI0 .DATA.L PWM5MN
35 ;
36 000070 36 .ORG H'000070
37 000070 00010000 37 IMIA1 .DATA.L PWM5MN
38 000074 00010000 38 IMIB1 .DATA.L PWM5MN
39 000078 00010000 39 OVI1 .DATA.L PWM5MN
40 ;
41 000080 41 .ORG H'000080
42 000080 00010000 42 IMIA2 .DATA.L PWM5MN
43 000084 00010000 43 IMIB2 .DATA.L PWM5MN
```

```

44 000088 00010000      44 OVI2      .DATA.L    PWM5MN
45
46 000090      46          .ORG      H'000090
47 000090 00010000      47 IMIA3     .DATA.L    PWM5MN
48 000094 00010000      48 IMIB3     .DATA.L    PWM5MN
49 000098 00010000      49 OVI3      .DATA.L    PWM5MN
50
51 0000A0      51          .ORG      H'0000A0
52 0000A0 00010000      52 IMIA4     .DATA.L    PWM5MN
53 0000A4 00010000      53 IMIB4     .DATA.L    PWM5MN
54 0000A8 00010000      54 OVI4      .DATA.L    PWM5MN
55
56 0000B0      56          .ORG      H'0000B0
57 0000B0 00010000      57 DEND0A    .DATA.L    PWM5MN
58 0000B4 00010000      58 DEND0B    .DATA.L    PWM5MN
59 0000B8 00010000      59 DEND1A    .DATA.L    PWM5MN
60 0000BC 00010000      60 DEND1B    .DATA.L    PWM5MN
61 0000C0 00010000      61 DEND2A    .DATA.L    PWM5MN
62 0000C4 00010000      62 DEND2B    .DATA.L    PWM5MN
63 0000C8 00010000      63 DEND3A    .DATA.L    PWM5MN
64 0000CC 00010000      64 DEND3B    .DATA.L    PWM5MN
65 0000D0 00010000      65 ERI0      .DATA.L    PWM5MN
66 0000D4 00010000      66 RXI0      .DATA.L    PWM5MN
67 0000D5 00010000      67 TXI0      .DATA.L    PWM5MN
68 0000DC 00010000      68 TEI0      .DATA.L    PWM5MN
69 0000E0 00010000      69 ERI1      .DATA.L    PWM5MN
70 0000E4 00010000      70 RXI1      .DATA.L    PWM5MN
71 0000E8 00010000      71 TXI1      .DATA.L    PWM5MN
72 0000EC 00010000      72 TEI1      .DATA.L    PWM5MN
73 0000F0 00010000      73 ADI       .DATA.L    PWM5MN
74
75 ;*****
76 ;*
77 ;*          RAM ALLOCATION
78 ;*
79 ;*****
80 ;
81 FFFFE0      81          .SECTION  RAM,DATA,LOCATE=H'FFFEF0
82
83 FFFFE0 00000002      83 PUL_CYC0  .RES.W    1          ;Pulse cycle time (ch0)
84 FFFFE2 00000002      84 PUL_SET0  .RES.W    1          ;Pulse cycle timing (ch0)
85
86 FFFFE4 00000002      86 PUL_CYC1  .RES.W    1          ;Pulse cycle time (ch1)
87 FFFFE6 00000002      87 PUL_SET1  .RES.W    1          ;Pulse set timing (ch1)
88
89 FFFFE8 00000002      89 PUL_CYC2  .RES.W    1          ;Pulse cycle time (ch2)

```

```

90 FFFEFA 00000002 90 PUL_SET2 .RES.W 1 ;Pulse set timing (ch2)
91 ;
92 FFFEFC 00000002 92 PUL_CYC3 .RES.W 1 ;Pulse cycle time (ch3)
93 FFFEFE 00000002 93 PUL_SET3 .RES.W 1 ;Pulse set timing (ch3)
94 ;
95 FFFF00 00000002 95 PUL_CYC4 .RES.W 1 ;Pulse cycle time (ch4)
96 FFFF02 00000002 96 PUL_SET4 .RES.W 1 ;Pulse set timing (ch4)
97 ;
98 ;*****
99 ;* *
100 ;* SYMBOL DEFINITIONS *
101 ;* *
102 ;*****
103 ;
104 00FFFF60 104 TSTR .EQU H'FFFF60 ;Timer start register
105 00FFFF61 105 TSNC .EQU H'FFFF61 ;Timer syncro register
106 00FFFF62 106 TMDR .EQU H'FFFF62 ;Timer mode register
107 ;
108 00FFFF64 108 TCR0 .EQU H'FFFF64 ;Timer control register0
109 00FFFF6A 109 GRA0 .EQU H'FFFF6A ;General register A0
110 00FFFF6C 110 GRB0 .EQU H'FFFF6C ;General register B0
111 ;
112 00FFFF6E 112 TCR1 .EQU H'FFFF6E ;Timer control register1
113 00FFFF74 113 GRA1 .EQU H'FFFF74 ;General register A1
114 00FFFF76 114 GRB1 .EQU H'FFFF76 ;General register B1
115 ;
116 00FFFF78 116 TCR2 .EQU H'FFFF78 ;Timer control register2
117 00FFFF7E 117 GRA2 .EQU H'FFFF7E ;General register A2
118 00FFFF80 118 GRB2 .EQU H'FFFF80 ;General register B2
119 ;
120 00FFFF82 120 TCR3 .EQU H'FFFF82 ;Timer control register3
121 00FFFF88 121 GRA3 .EQU H'FFFF88 ;General register A3
122 00FFFF8A 122 GRB3 .EQU H'FFFF8A ;General register B3
123 ;
124 00FFFF92 124 TCR4 .EQU H'FFFF92 ;Timer control register4
125 00FFFF98 125 GRA4 .EQU H'FFFF98 ;General register A4
126 00FFFF9A 126 GRB4 .EQU H'FFFF9A ;General register B4
127 ;
128 ;*****
129 ;* *
130 ;* MAIN PROGRAM : PWM5MN *
131 ;* *
132 ;*****
133 ;
134 010000 134 .SECTION PROG, CODE, LOCATE=H'010000
135 ;

```

136	00010000	136	PWM5MN:	.EQU	\$
137	010000	7A0700FFFEFE	137	MOV.L	#H'FFFEFE,SP ;Initialize stack pointer
138	010006	F8FF	138	MOV.B	#B'11111111,R0L
139	010008	3861	139	MOV.B	R0L,@TSNC ;Set synchronization ;mode ch0~4
140			140		
141	01000A	F8C0	141	MOV.B	#B'11000000,R0L
142	01000C	3864	142	MOV.B	R0L,@TCR0 ;Initialize TCR0
143	01000E	F8E0	143	MOV.B	#B'11100000,R0L
144	010010	386E	144	MOV.B	R0L,@TCR1 ;Initialize TCR1~4
145	010012	3878	145	MOV.B	R0L,@TCR2
146	010014	3882	146	MOV.B	R0L,@TCR3
147	010016	3892	147	MOV.B	R0L,@TCR4
148			148	;	
149	010018	6B00FEF2	149	MOV.W	@PUL-SET0,R0
150	01001C	6B80FF6A	150	MOV.W	R0,@GRA0 ;Set pulse set timing ;(ch0~4)
151	010020	6B00FEF6	151	MOV.W	@PUL_SET1,R0
152	010024	6B80FF74	152	MOV.W	R0,@GRA1
153	010028	6B00FEFA	153	MOV.W	@PUL_SET2,R0
154	01002C	6B80FF7E	154	MOV.W	R0,@GRA2
155	010030	6B00FEFE	155	MOV.W	@PUL_SET3,R0
156	010034	6B80FF88	156	MOV.W	R0,@GRA3
157	010038	6B00FF02	157	MOV.W	@PUL_SET4,R0
158	01003C	6B80FF98	158	MOV.W	R0,@GRA4
159			159		
160	010040	6B00FEF0	160	MOV.W	@PUL_CYC0,R0
161	010044	6B80FF6C	161	MOV.W	R0,@GRB0 ;Set pulse cycle time ;(ch0~4)
162	010048	6B00FEF4	162	MOV.W	@PUL_CYC1,R0
163	01004C	6B80FF76	163	MOV.W	R0,@GRB1
164	010050	6B00FEF8	164	MOV.W	@PUL_CYC2,R0
165	010054	6B80FF80	165	MOV.W	R0@GRB2
166	010058	6B00FEFC	166	MOV.W	@PUL_CYC3,R0
167	01005C	6B80FF8A	167	MOV.W	R0,@GRB3
168	010060	6B00FF00	168	MOV.W	@PUL_CYC4,R0
169	010064	6B80FF9A	169	MOV.W	R0,@GRB4
170			170		
171	010068	F89F	171	MOV.B	#B'10011111,R0L
172	01006A	3862	172	MOV.B	R0L,@TMDR ;Set PWM mode (ch0~4)
173			173		
174	01006C	F8FF	174	MOV.B	#B'11111111,R0L
175	01006E	3860	175	MOV.B	R0L,@TSTR ;Set TCNT start data ;(ch0~4)
176			176		
177	010070	40FE	177	PWM5MN99	BRA PWM5MN99

178	178	
179	179	.END
*****TOTAL ERRORS	0	
*****TOTAL WARNINGS	0	

2.8 Timer Counter Simultaneous Reset with External Signal

MCU: H8/3003

Function: ITU (sync/PWM mode)

2.8.1 Specifications

Four-channel pulse output synchronized with the external signal's falling edge (figure 2.37).

The delay time and pulse width from the external signal falling edge can be varied within the following ranges:

$$250 \text{ ns} \leq \text{delay time} < \text{external signal cycle} - \text{pulse width}$$

$$62.5 \text{ ns} \leq \text{pulse width} < \text{external signal cycle} - \text{delay time}$$

During 16 MHz operation, the external signal pulse cycle can be set from 320 ns to 4.09 ms.

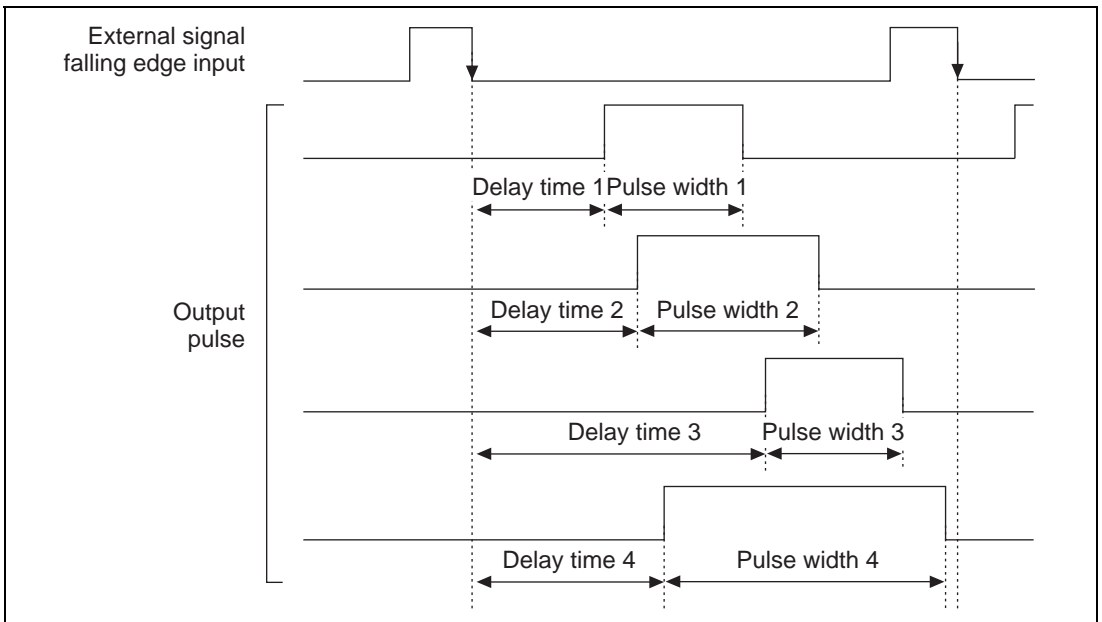


Figure 2.37 Example of Sync Pulse Output

2.8.2 Description of Functions Used

This sample task simultaneously resets multiple timer counters using an external signal, and provides 4-phase output.

Figure 2.38 shows a block diagram of the ITU/ch0 used in the sample task. The following functions are used by ch0 in external signal edge detection:

- Pulse falling edge detection function (input capture):
- Timer counter clear during pulse falling edge detection:

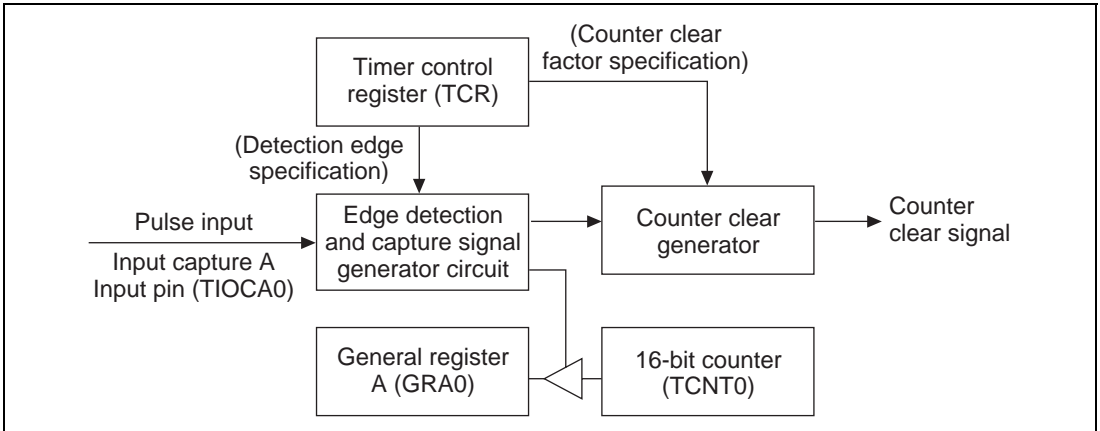


Figure 2.38 ITU/ch0 Block Diagram

Figure 2.39 shows a block diagram of the ITU/ch1–ch4 used in this sample task. The following functions are used in ch1–ch4 to synchronize with ch0 and provide 4-phase PWM waveform output:

- Simultaneous clearing of multiple timer counters (sync operation)
- Automatic switching of pulse output pin to PWM pin (PWM mode)

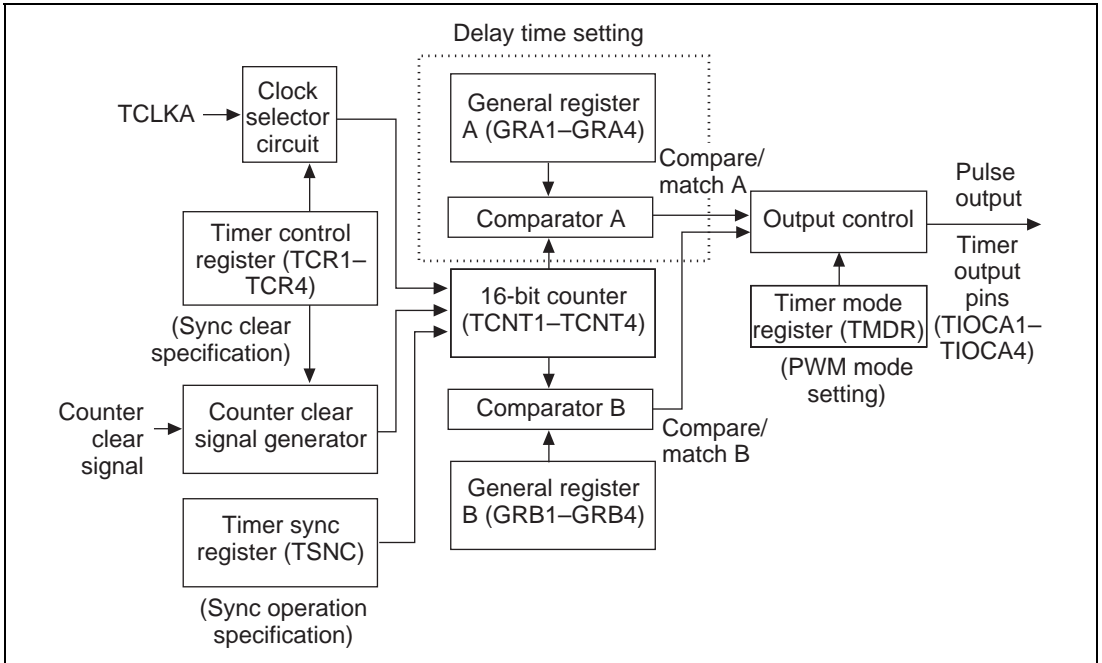


Figure 2.39 Sync Pulse Output Block Diagram

Table 2.37 shows the function allocation for this sample task. ITU functions are allocated, and delayed pulses are output to each channel at each falling edge input at ch0.

Table 2.37 ITU Function Allocation

ITU Function		Function
TSNC		Sets ch0–ch4 to sync operation
TMDR		Sets ch1–ch4 to PWM mode
TCLKA		External clock A pin for ch1–ch4
ch0	TCR0	Sets counter clear factor
	TIOCA0	Inputs reference pulse for output pulse
	GRA0	Detects counter value at external signal falling edge
ch1–ch4	TCR1–TCR4	Enables counter clock and counter clear factor
	TIOCA1–TIOCA4	Outputs pulse
	GRA1–GRA4	Sets delay time
	GRB1–GRB4	Sets timer value equivalent to delay time + pulse width

2.8.3 Description of Operations

One-Channel Pulse Output: Figure 2.40 shows the principle of operation. A PWM pulse is output from ch1 due to H8/3003 hardware and software processing.

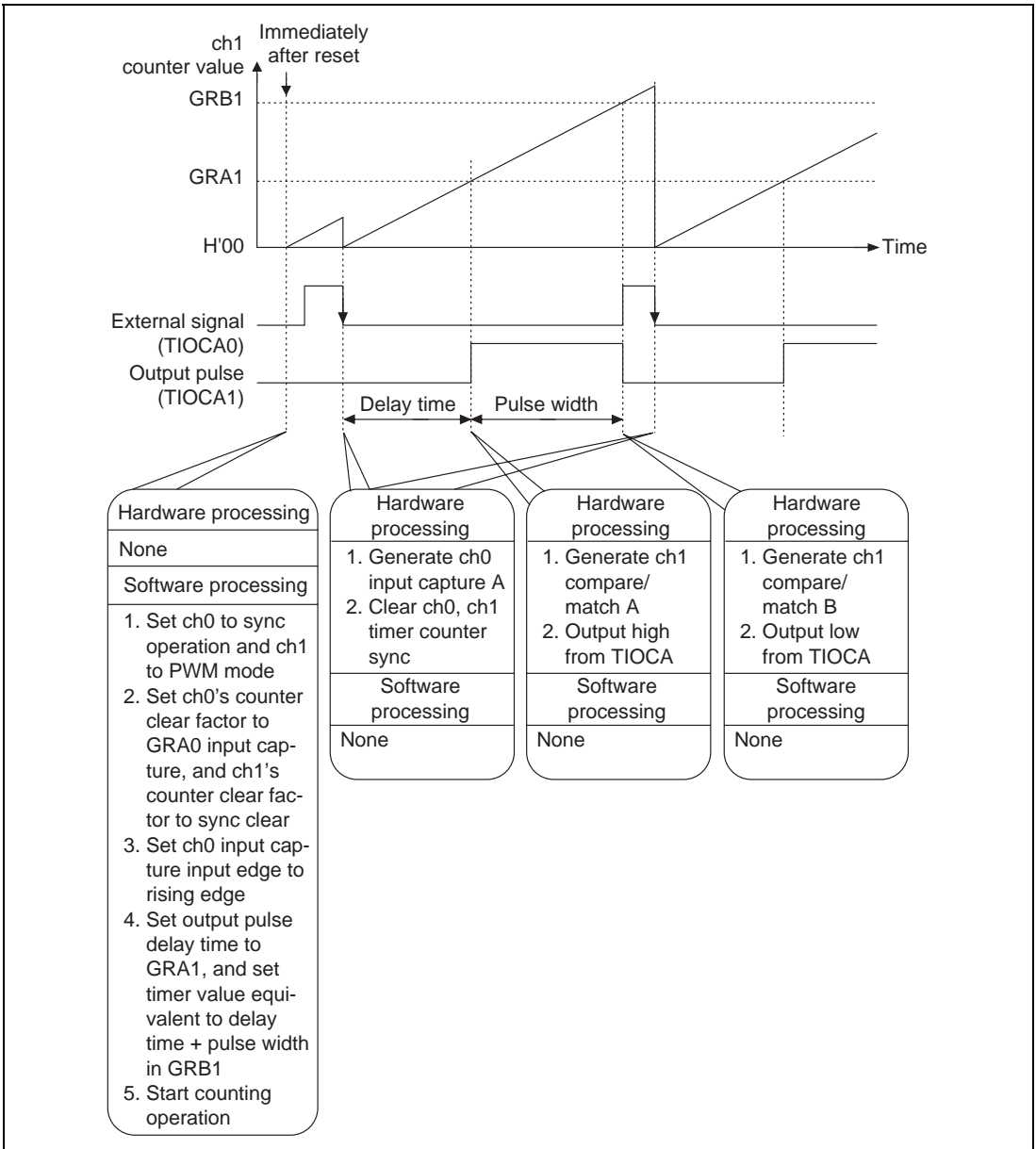


Figure 2.40 Operation Principle of Pulse Output

Four-Channel Pulse Output: Figure 2.41 shows the principle of operation. Duty pulses are output in sync with ch0 due to H8/3003 hardware and software processing.

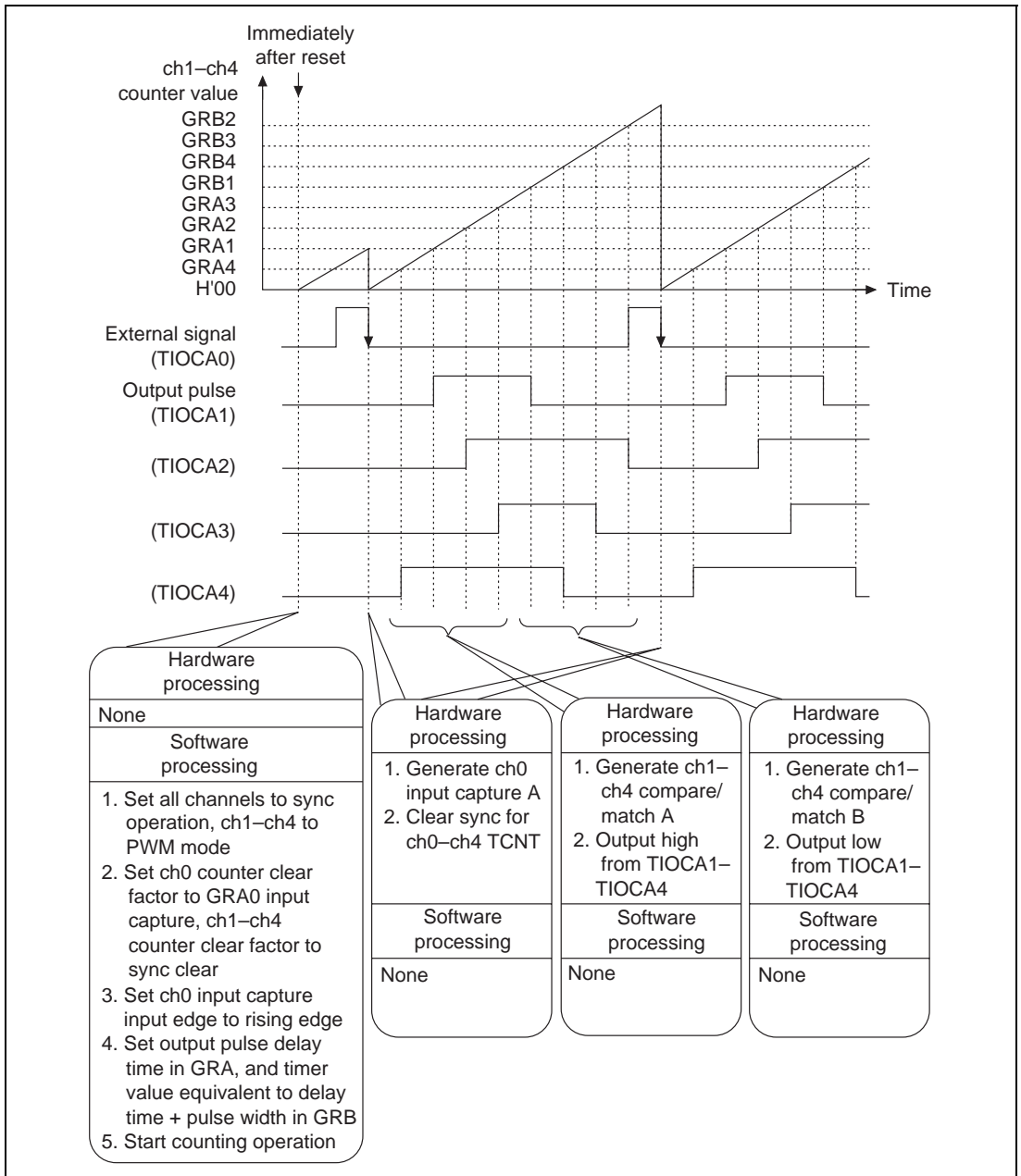


Figure 2.41 Operation Principle of 4-Channel Pulse Output

2.8.4 Description of Software

The software for timer counter simultaneous reset with external signal is described below.

Table 2.38 Description of Modules

Module Name	Label Name	Function
Main routine	CNTRSMN	Sets delay times and pulse widths in GRA1-4 and GRB1-4, and outputs 4-phase PWM pulses in sync with ch0.

Table 2.39 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
SET_DLY1– SET_DLY4	Sets timer value equivalent to output pulse delay time. Delay time is determined from the following formula: Delay time (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)	1 word	Main routine	Output
SET_WID1– SET_WID4	Sets timer value equivalent to output pulse delay time + pulse width. Pulse width is determined from the following formula Pulse width (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)	1 word	Main routine	Output

Table 2.40 Description of Internal Registers

Register Name		Function	Module Name
TSTR		Sets timer counter operation/stop	Main routine
TSNC		Enables/prohibits timer counter sync operation	
TMDR		Sets PWM mode enable/disable	
ch0	TCR	Set counter clear factor to input capture A	
	TIOR	Conducts transfer from TCNT to GRA during external signal falling edge detection, and simultaneously clears the counter	
ch1–ch4	TCR	Select input clock to TCNT and sets counter clear factor to sync clear	
	GRA	Sets output pulse delay time	
	GRB	Sets timer value equivalent to output pulse delay time + pulse width	

Table 2.41 Description of General Registers

Module Name	Register Name	Function
Main routine	R0	Used as work register during data setting.

Description of RAM: RAM not used other than for arguments in this sample task.

2.8.5 Flowchart

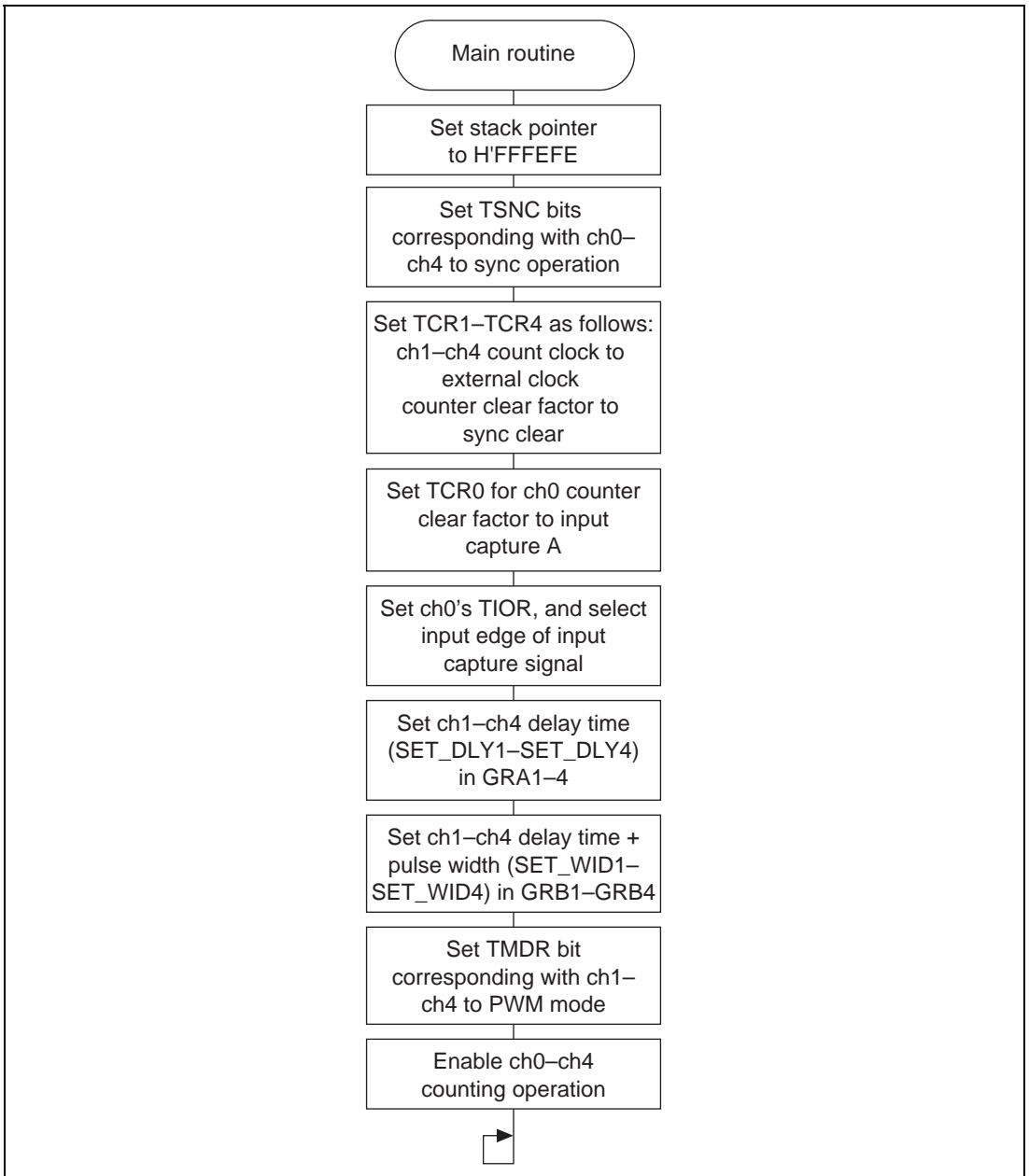


Figure 2.42 Main Routine Flowchart

2.8.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION  VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES      .DATA.L  CNTRSMN
10         10 ;
11 00001C   11         .ORG      H'00001C
12 00001C 00010000  12 NMI     .DATA.L  CNTRSMN
13 000020 00010000  13 TRAPA0  .DATA.L  CNTRSMN
14 000024 00010000  14 TRAPA1  .DATA.L  CNTRSMN
15 000028 00010000  15 TRAPA2  .DATA.L  CNTRSMN
16 00002C 00010000  16 TRAPA3  .DATA.L  CNTRSMN
17         17 ;
18 000030   18         .ORG      H'000030
19 000030 00010000  19 IRQ0    .DATA.L  CNTRSMN
20 000034 00010000  20 IRQ1    .DATA.L  CNTRSMN
21 000038 00010000  21 IRQ2    .DATA.L  CNTRSMN
22 00003C 00010000  22 IRQ3    .DATA.L  CNTRSMN
23 000040 00010000  23 IRQ4    .DATA.L  CNTRSMN
24 000044 00010000  24 IRQ5    .DATA.L  CNTRSMN
25 000048 00010000  25 IRQ6    .DATA.L  CNTRSMN
26 00004C 00010000  26 IRQ7    .DATA.L  CNTRSMN
27 000050 00010000  27 WOVI0   .DATA.L  CNTRSMN
28 000054 00010000  28 CMI     .DATA.L  CNTRSMN
29         29 ;
30 000060   30         .ORG      H'000060
31 000060 00010000  31 IMIA0   .DATA.L  CNTRSMN
32 000064 00010000  32 IMIB0   .DATA.L  CNTRSMN
33 000068 00010000  33 OVI0    .DATA.L  CNTRSMN
34         34 ;
35 000070   35         .ORG      H'000070
36 000070 00010000  36 IMIA1   .DATA.L  CNTRSMN
37 000074 00010000  37 IMIB1   .DATA.L  CNTRSMN
38 000078 00010000  38 OVI1    .DATA.L  CNTRSMN
39         39 ;
40 000080   40         .ORG      H'000080
41 000080 00010000  41 IMIA2   .DATA.L  CNTRSMN
42 000084 00010000  42 IMIB2   .DATA.L  CNTRSMN
43 000088 00010000  43 OVI2    .DATA.L  CNTRSMN

```

```

44                                     44 ;
45 000090                             45          .ORG          H'000090
46 000090 00010000                    46 IMIA3          .DATA.L    CNTRSMN
47 000094 00010000                    47 IMIB3          .DATA.L    CNTRSMN
48 000098 00010000                    48 OVI3          .DATA.L    CNTRSMN
49                                     49 ;
50 0000A0                             50          .ORG          H'0000A0
51 0000A0 00010000                    51 IMIA4          .DATA.L    CNTRSMN
52 0000A4 00010000                    52 IMIB4          .DATA.L    CNTRSMN
53 0000A8 00010000                    53 OVI4          .DATA.L    CNTRSMN
54                                     54 ;
55 0000B0                             55          .ORG          H'0000B0
56 0000B0 00010000                    56 DEND0A        .DATA.L    CNTRSMN
57 0000B4 00010000                    57 DEND0B        .DATA.L    CNTRSMN
58 0000B8 00010000                    58 DEND1A        .DATA.L    CNTRSMN
59 0000BC 00010000                    59 DEND1B        .DATA.L    CNTRSMN
60 0000C0 00010000                    60 DEND2A        .DATA.L    CNTRSMN
61 0000C4 00010000                    61 DEND2B        .DATA.L    CNTRSMN
62 0000C8 00010000                    62 DEND3A        .DATA.L    CNTRSMN
63 0000CC 00010000                    63 DEND3B        .DATA.L    CNTRSMN
64 0000D0 00010000                    64 ERI0          .DATA.L    CNTRSMN
65 0000D4 00010000                    65 RXI0          .DATA.L    CNTRSMN
66 0000D8 00010000                    66 TXI0          .DATA.L    CNTRSMN
67 0000DC 00010000                    67 TEI0          .DATA.L    CNTRSMN
68 0000E0 00010000                    68 ERI1          .DATA.L    CNTRSMN
69 0000E4 00010000                    69 RXI1          .DATA.L    CNTRSMN
70 0000E8 00010000                    70 TXI1          .DATA.L    CNTRSMN
71 0000EC 00010000                    71 TEI1          .DATA.L    CNTRSMN
72 0000F0 00010000                    72 ADI           .DATA.L    CNTRSMN
73                                     73 ;
74                                     74 ;*****
75                                     75 ;*                                     *
76                                     76 ;*          RAM ALLOCATION          *
77                                     77 ;*                                     *
78                                     78 ;*****
79                                     79 ;
80 FFFF00                             80          .SECTION    RAM,DATA,LOCATE=H'FFFF00
81                                     81 ;
82 FFFF00 00000002                    82 SET_DLY1      .RES.W    1          ;Delay time
83 FFFF02 00000002                    83 SET_WID1      .RES.W    1          ;Pulse width time
84                                     84 ;
85 FFFF04 00000002                    85 SET_DLY2      .RES.W    1          ;Delay time
86 FFFF06 00000002                    86 SET_WID2      .RES.W    1          ;Pulse width time
87                                     87 ;
88 FFFF0B 00000002                    88 SET_DLY3      .RES.W    1          ;Delay time
89 FFFF0A 00000002                    89 SET_WID3      .RES.W    1          ;Pulse width time

```

```

90          90 ;
91 FFFF0C 00000002 91 SET_DLY4 .RES.W 1 ;Delay time
92 FFFF0E 00000002 92 SET_WID4 .RES.W 1 ;Pulse width time
93          93 ;
94          94 ;*****
95          95 ;* *
96          96 ;* SYMBOL DEFINITIONS *
97          97 ;* *
98          98 ;*****
99          99 ;
100         00FFFF60 100 TSTR .EQU H'FFFF60 ;Timer start register
101         00FFFF61 101 TSNC .EQU H'FFFF61 ;Timer synchro register
102         00FFFF62 102 TMDR .EQU H'FFFF62 ;Timer mode register
103         103 ;
104         00FFFF64 104 TCR0 .EQU H'FFFF64 ;Timer control register0
105         00FFFF65 105 TIOR0 .EQU H'FFFF65 ;Timer I/O control
register0
106         106 ;
107         00FFFF6E 107 TCR1 .EQU H'FFFF6E ;Timer control register1
108         00FFFF74 108 GRA1 .EQU H'FFFF74 ;General register A1
109         00FFFF76 109 GRB1 .EQU H'FFFF76 ;General register B1
110         110 ;
111         00FFFF78 111 TCR2 .EQU H'FFFF78 ;Timer control register2
112         00FFFF7E 112 GRA2 .EQU H'FFFF7E ;General register A2
113         00FFFF80 113 GRB2 .EQU H'FFFF80 ;General register B2
114         114 ;
115         00FFFF82 115 TCR3 .EQU H'FFFF82 ;Timer control Register3
116         00FFFF88 116 GRA3 .EQU H'FFFF88 ;General register A3
117         00FFFF8A 117 GRB3 .EQU H'FFFF8A ;General register B3
118         118 ;
119         00FFFF92 119 TCR4 .EQU H'FFFF92 ;Timer control Register4
120         00FFFF98 120 GRA4 .EQU H'FFFF98 ;General register A4
121         00FFFF9A 121 GRB4 .EQU H'FFFF9A ;General register B4
122         122 ;
123         123 ;*****
124         124 ;* *
125         125 ;* MAIN PROGRAM : CNTRSMN *
126         126 ;* *
127         127 ;*****
128         128 ;
129 010000 129 .SECTION PROG, CODE, LOCATE=H'010000
130         130 ;
131         00010000 131 CNTRSMN: .EQU $
132 010000 7A0700FFFEFE 132 MOV.L #H'FFFEFE, SP ;Initialize stack pointer
133 010006 F8FF 133 MOV.B #B'11111111, R0L
134 010008 3861 134 MOV.B R0L, @TSNC ;Initialize TSNC

```

135		135		
136	01000A F8A0	136	MOV.B	#B'10100000,R0L
137	01000C 3864	137	MOV.B	R0L,@TCR0 ;Initialize TCR0
138	01000E F8E0	138	MOV.B	#B'11100000,R0L
139	010010 386E	139	MOV.B	R0L,@TCR1 ;Initialize TCR1~4
140	010012 3878	140	MOV.B	R0L,@TCR2
141	010014 3882	141	MOV.B	R0L,@TCR3
142	010016 3892	142	MOV.B	R0L,@TCR4
143		143 ;		
144	010018 F88D	144	MOV.B	#B'10001101,R0L
145	01001A 3865	145	MOV.B	R0L,@TIOR0 ;Initialize TIOR0
146		146		
147	01001C 6B00FF00	147	MOV.W	@SET_DLY1,R0
148	010020 6B80FF74	148	MOV.W	R0,@GRA1 ;Set delay timel~4
149	010024 6B00FF04	149	MOV.W	@SET_DLY2,R0
150	010028 6B80FF7E	150	MOV.W	R0,@GRA2
151	01002C 6B00FF06	151	MOV.W	@SET_DLY3,R0
152	010030 6B80FF88	152	MOV.W	R0,@GRA3
153	010034 6B00FF0C	153	MOV.W	@SET_DLY4,R0
154	010038 6B80FF98	154	MOV.W	R0,@GRA4
155		155		
156	01003C 6B00FF02	156	MOV.W	@SET_WID1,R0
157	010040 6B80FF76	157	MOV.W	R0,@GRB1 ;Set width1~4
158	010044 6B00FF06	158	MOV.W	@SET_WID2,R0
159	010048 6B80FF80	159	MOV.W	R0,@GRB2
160	01004C 6B00FF0A	160	MOV.W	@SET_WID3,R0
161	010050 6B80FF8A	161	MOV.W	R0,@GRB3
162	010054 6B00FF0E	162	MOV.W	@SET_WID4,R0
163	010058 6B80FF9A	163	MOV.W	R0,@GRB4
164		164		
165	01005C F89E	165	MOV.B	#B'10011110,R0L
166	01005E 3862	166	MOV.B	R0L,@TMDR ;Set PWM mode chl~4
167	010060 F8FE	167	MOV.B	#B'11111110,R0L
168	010062 3860	168	MOV.B	R0L,@TSTR ;Start TCNT1~4
169		169		
170	010064 40FE	170	CNTRS99 BRA	CNTRS99
171		171		
172		172	.END	
****	TOTAL ERRORS	0		
****	TOTAL WARNINGS	0		

2.9 One-Shot Pulse Output Using Buffer Mode

MCU: H8/3003

Function ITU: (buffer operation)

2.9.1 Specifications

Output of one-shot pulse synchronized with the external signal's falling edge (figure 2.43)

The delay time and pulse width from the external signal falling edge can be varied within the following ranges:

$$2 \mu\text{s} \leq \text{delay time} < \text{reference pulse cycle} - \text{pulse width}$$

$$62.5 \text{ ns} \leq \text{pulse width} < \text{reference pulse cycle} - \text{delay time}$$

A reference pulse frequency can be input from 400 KHz.

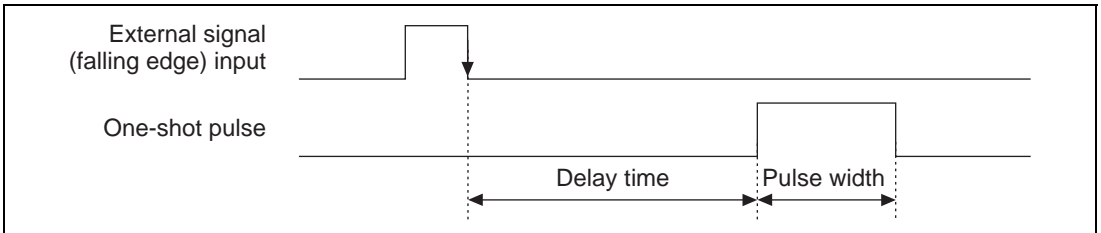


Figure 2.43 One-Shot Pulse Output

2.9.2 Description of Functions Used

In this sample task, a one-shot pulse is output using Group 0 DMAC/ch0A, ch0B, and ITU ch0 and ch3. Figure 2.44 shows a block diagram of ITU/ch3 used in this sample task.

The following ITU/ch3 functions are used to output one-shot pulses:

- Transfer of buffer register contents to general registers during compare/match (buffer operation)
- Automatic pulse output function in hardware without software intervention (output compare)
- Reversal of output with each compare/match (toggle output)
- Simultaneous clearing of multiple timer counters during input capture (sync clear)

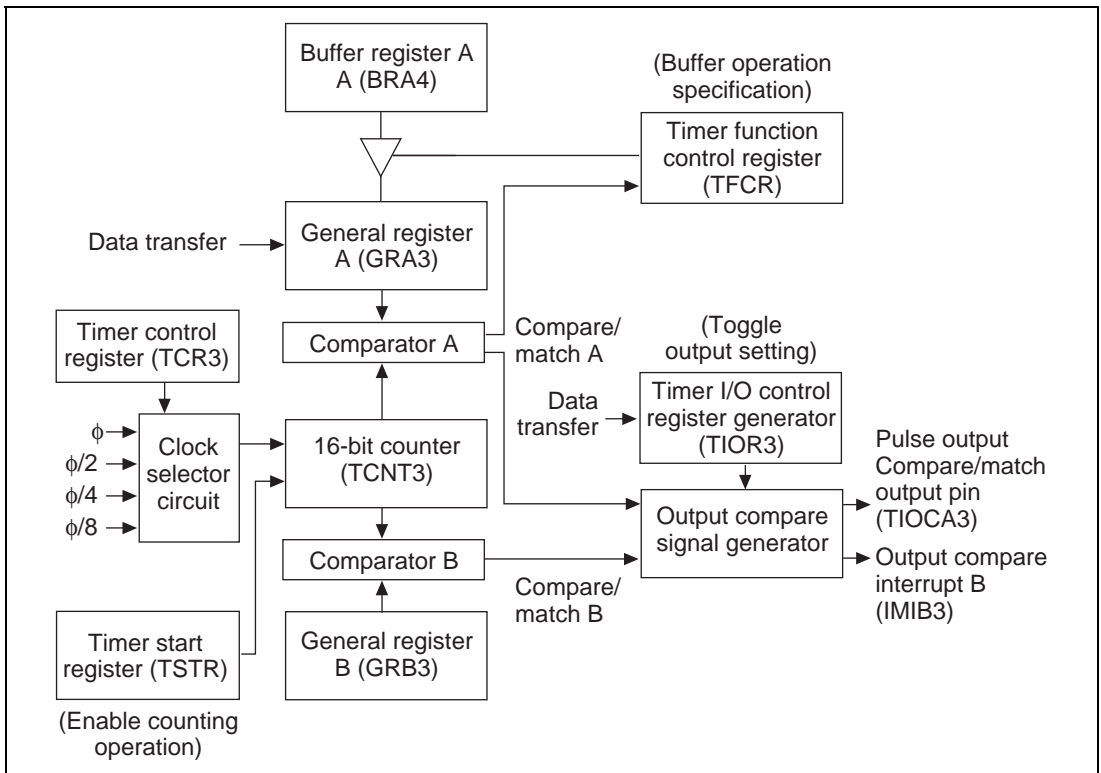


Figure 2.44 ITU/ch3 Block Diagram

Figure 2.45 shows a block diagram of the DMAC and ITU used in this sample task. One-shot pulses are output using the following DMAC and ITU function:

- DMAC startup during ITU input capture

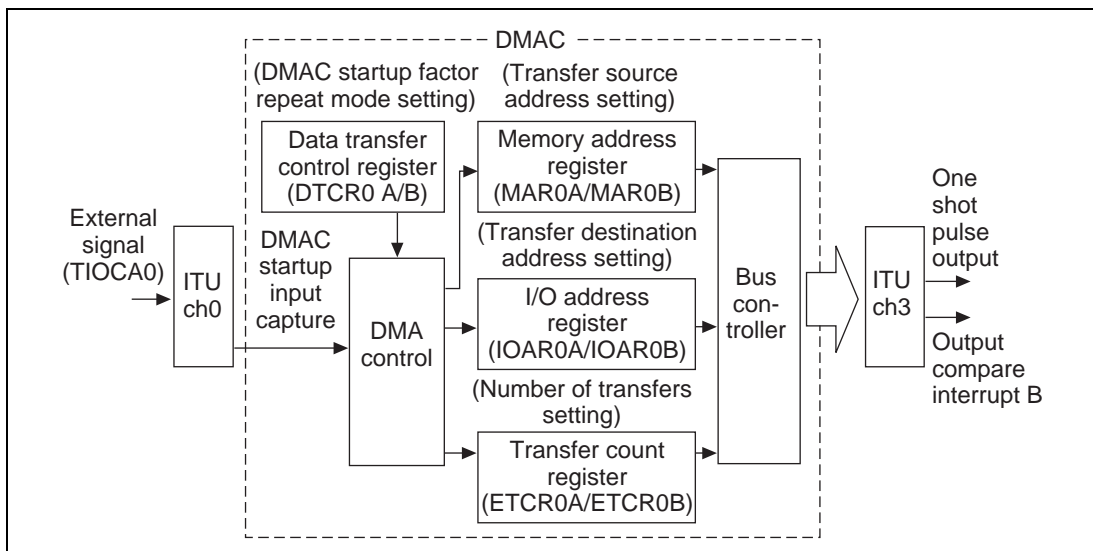


Figure 2.45 One-Shot Output Block Diagram

Table 2.42 shows the function allocation in this sample task. H8/300H functions are allocated for one-shot pulse output.

Table 2.42 H8/300H Function Allocation

H8/300H On-chip Functions		Function	
ITU	TFCR	Sets buffer operation	
	TSTR	Selects timer counter operation/stop	
	ch0	TCR0	Sets counter clear factor
		TIOR0	Selects input capture function
		TIOCA0	Inputs reference pulse for one-shot pulse output
	ch3	TCR3	Selects TCNT3 input clock and counter clear factor
		TIOR3	Sets output level from TIOCA3 during compare/match
		TIER3	Enables/disables interrupt requests due to IMFB flag
		TSR3	Indicates occurrence of compare/match
		GRA3	Sets one-shot pulse delay time
		BRA3	Sets one-shot pulse reset timing
		GRB3	Sets one-shot pulse output prohibit timing
		TIOCA3	Outputs one-shot pulse
	DMAC	ch0 A/B	DTCR0 A/B
MAR0 A/B			Sets addresses of data for transfer in each register
IORA0 A/B			Sets transfer destination address for each channel
ETCR0 A/B			Sets number of transfers for each channel

2.9.3 Description of Operations

Figure 2.46 shows the principle of operation. One-shot pulses are output due to H8/3003 hardware and software processing.

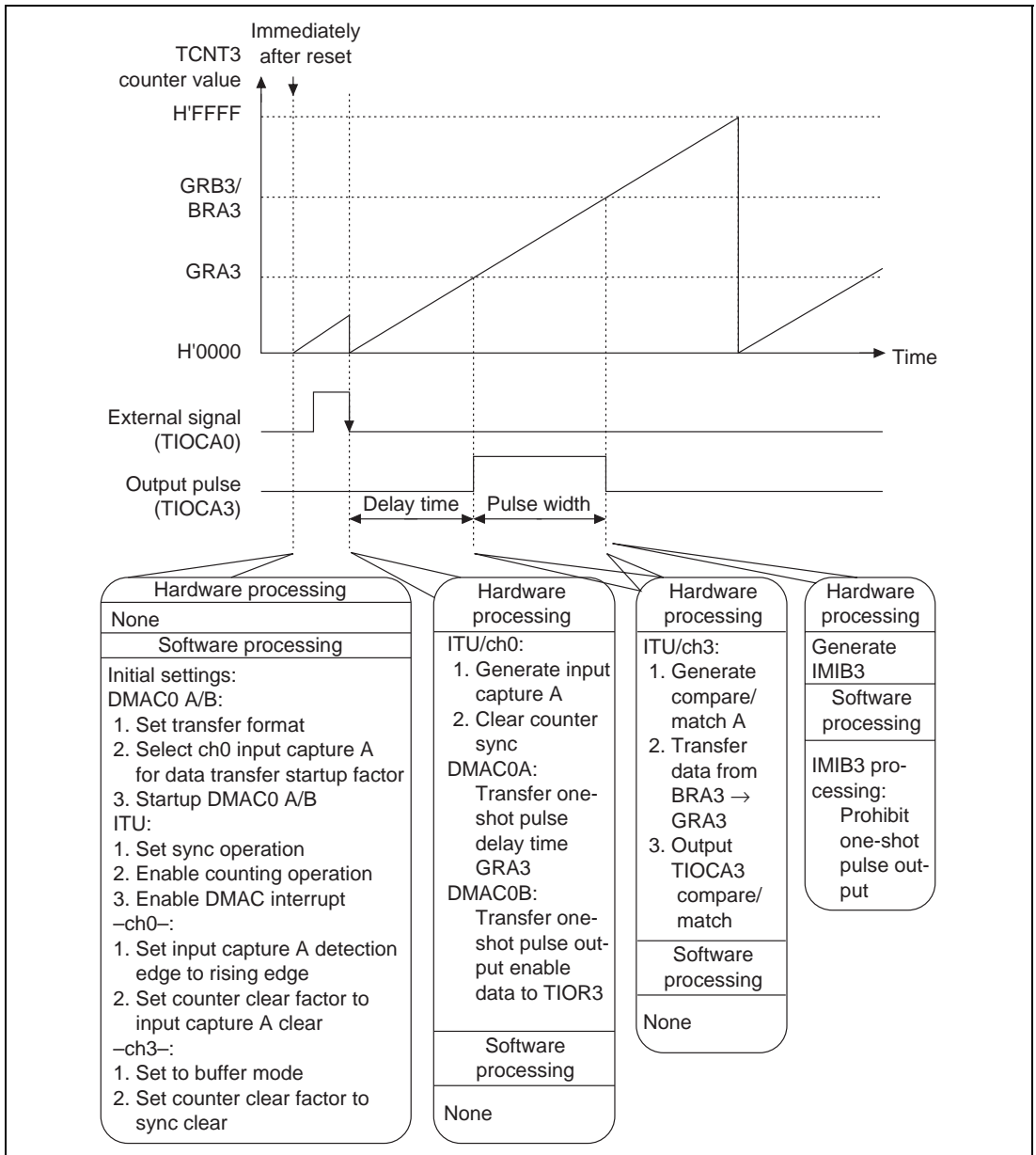


Figure 2.46 Principle of One-Shot Pulse Output Operation

2.9.4 Description of Software

The software for one-shot pulse output using buffer mode is described below.

Table 2.43 Description of Modules

Module Name	Label Name	Function
Main routine	ONEMN	Sets delay time and pulse width in GRA3 and BRA3, and provides one-shot pulse output
Pulse output prohibit	POUTDLE	Prohibits pulse output

Table 2.44 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
SET_DLY	Sets timer value equivalent to one-shot pulse delay time Delay time is determined from the following formula: Delay time (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)	1 word	Main routine	Input
ONE_RST	Sets timer value equivalent to one-shot pulse reset timing Reset timing is determined from the following formula: Pulse reset timing (ns) = timer value × ϕ cycle (62.5 ns during 16 MHz operation)	1 word	Main routine	Input
IO_CNTR	Sets one-shot pulse output enable data	1 word	Main routine	Output

Table 2.45 Description of Internal Registers

Register Name		Function	Module Name	
ITU	TSTR	Selects timer counter operation/stop	Main routine	
	TFCR	Sets GRA3/BRA3 to buffer operation		
	TSNC	Sets sync operation		
	ch0	TCR0	Sets counter clear factor	
		TIOR0	Detects input pulse falling edge	
	ch3	TCR3	Sets clock input to TCNT and counter clear factor	
		TIOR3	Sets output level from TIOCA3 during compare/match A	Pulse output prohibit
		TIER3	Enables interrupts due to IMFB	Main routine/ pulse output prohibit
		TSR3	Indicates compare/match occurrence due to GRB	Main routine
		GRA3	Sets one-shot pulse delay time	
		BRA3	Sets one-shot pulse reset timing	
		GRB3	Sets one-shot pulse output prohibit timing	
	DMAC	ch0 A/B	DTCR0	Sets operation of each DMAC channel
MAR0			Sets addresses of data for transfer in each register	
IOAR0			Sets transfer destination address for each channel	
ETCR0			Sets number of transfers for each channel	

Table 2.46 Description of General Registers

Module Name	Register Name	Function
Main routine	ER0	Used as work register during data setting
Pulse output prohibit	ROL	Used as work register during data setting

Description of RAM: RAM not used other than for arguments in this sample task.

2.9.5 Flowcharts

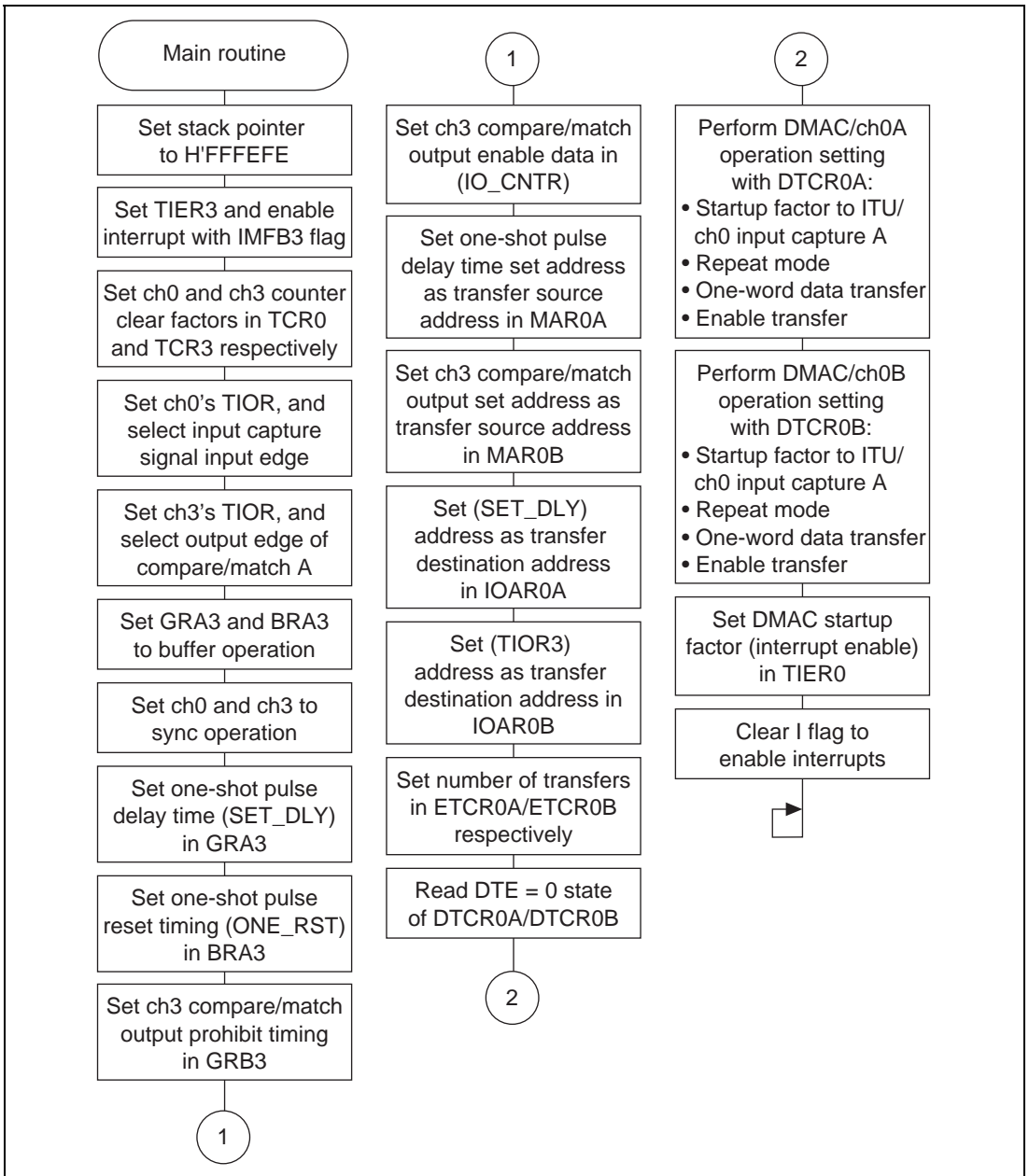


Figure 2.47 Main Routine Flowchart

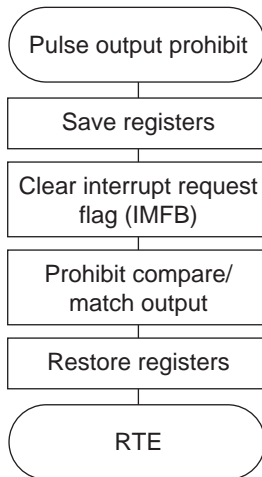


Figure 2.48 Pulse Output Prohibit Flowchart

2.9.6 Program List

```
1 ;*****
2 ;*
3 ;* VECTOR ADDRESS
4 ;*
5 ;*****
6 ;
7 .CPU 300HA
8 000000 8 ; .SECTION VECT, CODE, LOCATE=H'000000
9 ;
10 000000 00010000 10 RES .DATA.L ONEMN
11 ;
12 00001C 12 .ORG H'00001C
13 00001C 00010000 13 NMI .DATA.L ONEMN
14 000020 00010000 14 TRAPA0 .DATA.L ONEMN
15 000024 00010000 15 TRAPA1 .DATA.L ONEMN
16 000028 00010000 16 TRAPA2 .DATA.L ONEMN
17 00002C 00010000 17 TRAPA3 .DATA.L ONEMN
18 ;
19 000030 19 .ORG H'000030
20 000030 00010000 20 IRQ0 .DATA.L ONEMN
21 000034 00010000 21 IRQ1 .DATA.L ONEMN
22 000038 00010000 22 IRQ2 .DATA.L ONEMN
23 00003C 00010000 23 IRQ3 .DATA.L ONEMN
24 000040 00010000 24 IRQ4 .DATA.L ONEMN
25 000044 00010000 25 IRQ5 .DATA.L ONEMN
26 000048 00010000 26 IRQ6 .DATA.L ONEMN
27 00004C 00010000 27 IRQ7 .DATA.L ONEMN
28 000050 00010000 28 WOVI0 .DATA.L ONEMN
29 000054 00010000 29 CMI .DATA.L ONEMN
30 ;
31 000060 31 .ORG H'000060
32 000060 00010000 32 IMIA0 .DATA.L ONEMN
33 000064 00010000 33 IMIB0 .DATA.L ONEMN
34 000068 00010000 34 OVI0 .DATA.L ONEMN
35 ;
36 000070 36 .ORG H'000070
37 000070 00010000 37 IMIA1 .DATA.L ONEMN
38 000074 00010000 38 IMIB1 .DATA.L ONEMN
39 000078 00010000 39 OVI1 .DATA.L ONEMN
40 ;
41 000080 41 .ORG H'000080
42 000080 00010000 42 IMIA2 .DATA.L ONEMN
43 000084 00010000 43 IMIB2 .DATA.L ONEMN
```

```

44 000088 00010000      44 OVI2      .DATA.L  ONEMN
45                               45 ;
46 000090                               46          .ORG      H'000090
47 000090 00010000      47 IMIA3     .DATA.L  ONEMN
48 000094 0001007A      48 IMIB3     .DATA.L  POUTDLE
49 000098 00010000      49 OVI3      .DATA.L  ONEMN
50                               50 ;
51 0000A0                               51          .ORG      H'0000A0
52 0000A0 00010000      52 IMIA4     .DATA.L  ONEMN
53 0000A4 00010000      53 IMIB4     .DATA.L  ONEMN
54 0000A8 00010000      54 OVI4      .DATA.L  ONEMN
55                               55 ;
56 0000B0                               56          .ORG      H'0000B0
57 0000B0 00010000      57 DEND0A    .DATA.L  ONEMN
58 0000B4 00010000      58 DEND0B    .DATA.L  ONEMN
59 0000B8 00010000      59 DEND1A    .DATA.L  ONEMN
60 0000BC 00010000      60 DEND1B    .DATA.L  ONEMN
61 0000C0 00010000      61 DEND2A    .DATA.L  ONEMN
62 0000C4 00010000      62 DEND2B    .DATA.L  ONEMN
63 0000C8 00010000      63 DEND3A    .DATA.L  ONEMN
64 0000CC 00010000      64 DEND3B    .DATA.L  ONEMN
65 0000D0 00010000      65 ERI0      .DATA.L  ONEMN
66 0000D4 00010000      66 RXI0      .DATA.L  ONEMN
67 0000D8 00010000      67 TXI0      .DATA.L  ONEMN
68 0000DC 00010000      68 TEI0      .DATA.L  ONEMN
69 0000E0 00010000      69 ERI1      .DATA.L  ONEMN
70 0000E4 00010000      70 RXI1      .DATA.L  ONEMN
71 0000E8 00010000      71 TXI1      .DATA.L  ONEMN
72 0000EC 00010000      72 TEI1      .DATA.L  ONEMN
73 0000F0 00010000      73 ADI       .DATA.L  ONEMN
74                               74 ;
75                               75 ;*****
76                               76 ;*                                     *
77                               77 ;*          RAM ALLOCATION          *
78                               78 ;*                                     *
79                               79 ;*****
80                               80 ;
81 FFFF00                               81          .SECTION  RAM,DATA,LOCATE=H'FFFF00
82                               82 ;
83 FFFF00 00000002      83 SET_DLY   .RES.W   1          ;Pulse delay time
84 FFFF02 00000002      84 ONE_RST   .RES.W   1          ;Pulse reset timing
85 FFFF04 00000001      85 IO_CNTR   .RES.B   1          ;I/O control
86                               86 ;
87                               87 ;*****
88                               88 ;*                                     *
89                               89 ;*          SYMBOL DEFINITIONS          *

```



```

90          90 ;*
91          91 ;*****
92          92 ;
93          00FFFF20 93 MAR0A .EQU H'FFFF20 ;Memory address register 0A
94          00FFFF24 94 ETCR0A .EQU H'FFFF24 ;Execute transfer control
          ;register 0A
95          00FFFF26 95 IOAR0A .EQU H'FFFF26 ;I/O address register 0A
96          00FFFF27 96 DTCR0A .EQU H'FFFF27 ;Data transfer control
          ;register 0A
97          97 ;
98          00FFFF28 98 MAR0B .EQU H'FFFF28 ;Memory address register 0B
99          00FFFF2C 99 ETCR0B .EQU H'FFFF2C ;Execute transfer control
          ;register 0B
100         00FFFF2E 100 IOAR0B .EQU H'FFFF2E ;I/O address register 0B
101         00FFFF2F 101 DTCR0B .EQU H'FFFF2F ;Data transfer control
          ;register 0B
102         102 ;
103         103 ;
104         00FFFF60 104 TSTR .EQU H'FFFF60 ;Timer start register
105         00FFFF61 105 TSNC .EQU H'FFFF61 ;Timer sync register
106         00FFFF63 106 TFCR .EQU H'FFFF63 ;Timer function control
          ;register
107         107 ;
108         00FFFF64 108 TCR0 .EQU H'FFFF64 ;Timer control register0
109         00FFFF65 109 TIOR0 .EQU H'FFFF65 ;Timer I/O control register0
110        00FFFF66 110 TIER0 .EQU H'FFFF66 ;Timer interrupt enable
          ;register
111        00000000 111 IMIEA .EQU 0
112        112 ;
113        00FFFF82 113 TCR3 .EQU H'FFFF82 ;Timer control register3
114        00FFFF83 114 TIOR3 .EQU H'FFFF83 ;Timer I/O control register3
115        00FFFF84 115 TIER3 .EQU H'FFFF84 ;Timer interrupt enable
          ;register3
116        00FFFF85 116 TSR3 .EQU H'FFFF85 ;Timer status register3
117        00000001 117 IMFB3 .EQU 1 ;Inputcapture flag3
118        00FFFF86 118 TCNT3 .EQU H'FFFF86 ;Timer counter3
119        00FFFF88 119 GRA3 .EQU H'FFFF88 ;General register A3
120        00FFFF8A 120 GRB3 .EQU H'FFFF8A ;General register B3
121        00FFFF8C 121 BRA3 .EQU H'FFFF8C ;Buffer register A3
122        122 ;
123        123 ;*****
124        124 ;*
125        125 ;* MAIN PROGRAM : ONEMN *
126        126 ;*
127        127 ;*****
128        128 ;

```

```

129 010000          129          .SECTION  PROG, CODE, LOCATE=H'010000
130                130 ;
131          00010000 131 ONEMN:  .EQU      $
132 010000 7A0700FFFE 132          MOV.L      #H'FFFEFE, SP ;Initialize stack pointer
133                133 ;
134                134 ;-----ITU INITIALIZE-----
135                135 ;
136 010006 F8FA      136          MOV.B      #B'11111010, R0L
137 010008 3884      137          MOV.B      R0L, @TIER3 ;Initialize TIER3(1->IMIEB)
138 01000A F8A0      138          MOV.B      #B'10100000, R0L
139 01000C 3864      139          MOV.B      R0L, @TCR0 ;Initialize TCR0
140 01000E F8E0      140          MOV.B      #B'11100000, R0L
141 010010 3882      141          MOV.B      R0L, @TCR3 ;Initialize TCR3
142                142
143 010012 F88D      143          MOV.B      #B'10001101, R0L
144 010014 3865      144          MOV.B      R0L, @TIOR0 ;Initialize TIOR0
145 010016 F8C1      145          MOV.B      #B'11000001, R0L
146 010018 3863      146          MOV.B      R0L, @TFCR ;Initialize TFCR
147                147
148 01001A F8E9      148          MOV.B      #B'11101001, R0L
149 01001C 3861      149          MOV.B      R0L, @TSNC ;Initialize TSNC
150                150
151 01001E 6B00FF00 151          MOV.W      @SET_DLY, R0
152 010022 6B80FF88 152          MOV.W      R0, @GRA3 ;Set delay time
153 010026 6B00FF02 153          MOV.W      @ONE_RST, R0
154 01002A 6B80FF8C 154          MOV.W      R0, @BRA3 ;Set pulse reset timing
155 01002E 6B80FF8A 155          MOV.W      R0, @GRB3 ;Set pulse output disable
                                ;time
156 010032 F88B      156          MOV.B      #B'10001011, R0L
157 010034 3804      157          MOV.B      R0L, @IO_CNTR ;Set pulse output enable
                                ;data
158                158
159                159 ;
160                160 ;-----DMAC INITIALIZE-----
161                161 ;
162 010036 7A0000FFFF 162          MOV.L      #SET_DLY, ER0
163 01003C 01006B80FF20 163          MOV.L      ER0, @MAR0A ;Set base address
164 010042 7A0000FFFF04 164          MOV.L      #IO_CNTR, ER0
165 010048 01006B80FF28 165          MOV.L      ER0, @MAR0B ;Set base address
166                166
167 01004E F888      167          MOV.B      #H'88, R0L
168 010050 3826      168          MOV.B      R0L, @IOAR0A ;Set execute address
169 010052 F883      169          MOV.B      #H'83, R0L
170 010054 382E      170          MOV.B      R0L, @IOAR0B ;Set execute address
171                171
172 010056 79000101 172          MOV.W      #H'0101, R0

```

```

173 01005A 6B80FF24      173      MOV.W      R0,@ETCR0A      ;Set execute count
174 01005E 6B80FF2C      174      MOV.W      R0,@ETCR0B      ;Set execute count
175
176 010062 2827          176      MOV.B      @DTCR0A,R0L
177 010064 282F          177      MOV.B      @DTCR0B,R0L
178
179 010066 F8D0          179      MOV.B      #B'11010000,R0L
180 010068 3827          180      MOV.B      R0L,@DTCR0A      ;Set 0Achannel control
                                ;data
181 01006A F890          181      MOV.B      #B'10010000,R0L
182 01006C 382F          182      MOV.B      R0L,@DTCR0B      ;Set 0Bchannel control
                                ;data
183
184 01006E F8E9          184      MOV.B      #B'11101001,R0L
185 010070 3860          185      MOV.B      R0L,@TSTR      ;Start TCNT1,3
186
187 010072 7F667000      187      BSET      #IMIEA,@TIER0      ;Enable DMAC
188
189 010076 0700          189      LDC.B      #0,CCR          ;Enable interrupt
190
191 010078 40FE          191  ONEMN99  BRA      ONEMN99
192
193 ;*****
194 ;*
195 ;*          NAME : POUTDLE(PULSE OUTUPT DISABLE)
196 ;*
197 ;*****
198 ;*
199 ;*          ENTRY : NOTHING
200 ;*          RETURNS : TIOR3(TIMER I/O REG 3)
201 ;*
202 ;*****
203 ;
204          0001007A      204  POUTDLE  .EQU      $
205 01007A 01006DF0      205      PUSH.L      ER0          ;Escape register
206 01007E 7F857210      206      BCLR      #IMFB3,@TSR3      ;Clear IMFB3 request
                                ;(0->IMFB3)
207 010082 F889          207      MOV.B      #B'10001001,R0L
208 010084 3883          208      MOV.B      R0L,@TIOR3      ;Set pulse output
                                ;disable data
209 010086 01006D70      209      POP.L      ER0          ;Return register
210 01008A 5670          210      RTE
211
212
213          .END

```

```

****TOTAL ERRORS      0

```


2.10 4-bit × 4 System Output

MCU: H8/3003

Function: ITU (4-bit × 4 system output)

2.10.1 Specifications

- Asynchronous pulse output using TPC output
- ITU compare/match is used for the TPC startup factor

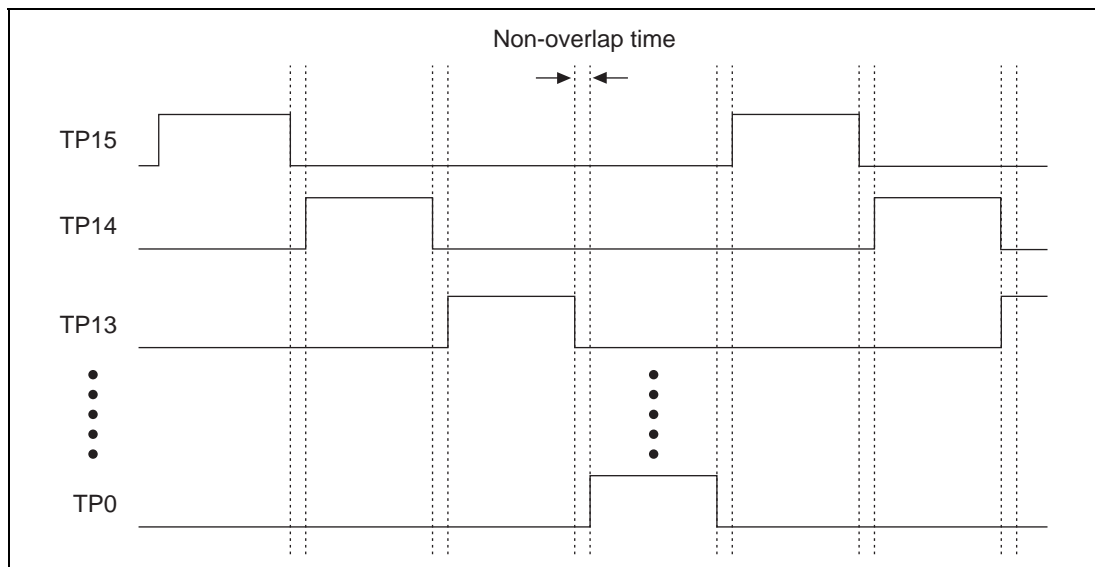


Figure 2.49 Output Example of 4-Bit × 4 System

2.10.2 Description of Functions Used

In this sample task, the ITU 4-channel 4-bit unit TPC output groups 3–0 are used for a 4 bit × 4 system of asynchronous pulse output. Figure 2.50 shows a block diagram of the TPC used in the sample task. The following functions are employed:

- With 4-bit unit group and output trigger signal selectable, a maximum 4 bit × 4 system output can be achieved
- Output trigger signal selectable for each group from the compare/match signals of the 4-channel ITU
- Non-overlap time among multiple pulse outputs can be set

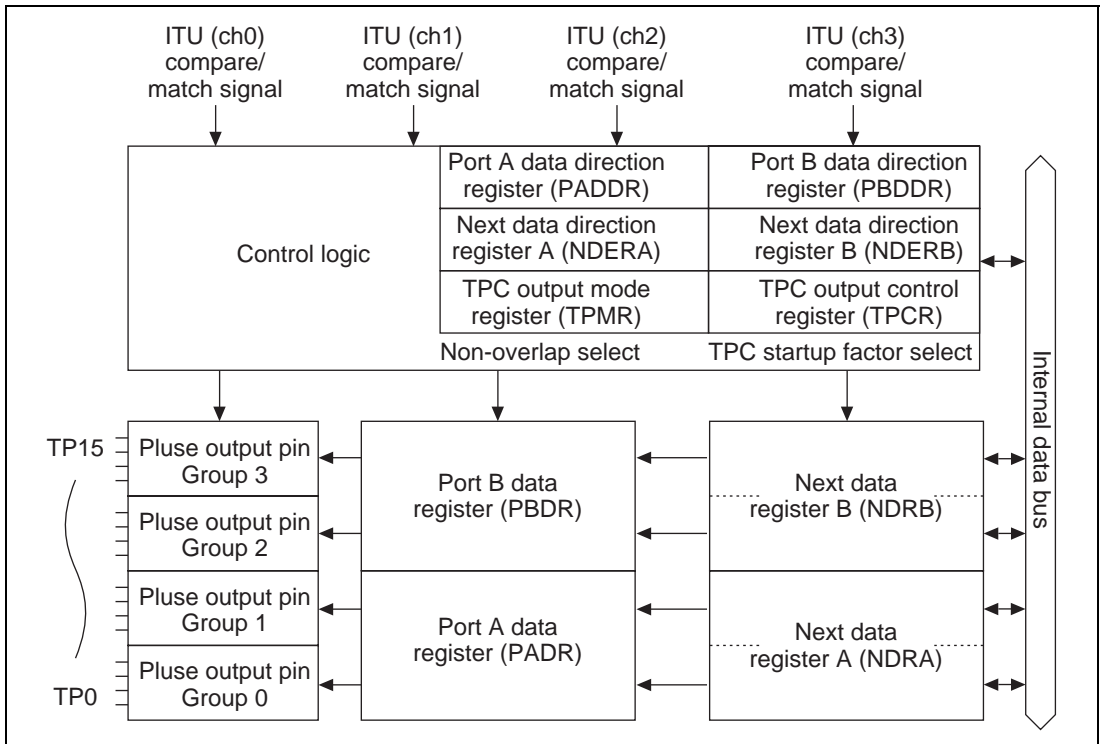


Figure 2.50 Timing Pattern Controller Block Diagram

Table 2.47 shows the function allocation for this sample task. ITU and TPC functions are allocated to provide 4 bit × 4 system output.

Table 2.47 TPC Function Allocation

Onchip Peripheral Function	Onchip Function	Function
TPC	TP15–TP0	Conducts 4 bit × 4 system output
	TPMR	Sets non-overlap mode
	TPCR	Sets output trigger signal for TPC output
	NDERB	Enables TP15–TP8 TPC output
	NDERA	Enables TP7–TP0 TPC output
	NDRA	Stores next TPC output data
	NDRB	Stores next TPC output data
	NDRC	Stores next TPC output data
	NDRD	Stores next TPC output data
	PADDR	Sets TPC output pin
	PBDDR	Sets TPC output pin
	PADR	Stores TP7–TP0 output data
	PBDR	Stores TP15–TP8 output data
	ITU	GRA0–GRA3
GRB0–GRB3		Sets TPC output trigger cycle
TCR0–TCR3		Sets counter clock and counter clear factor
TSR0–TSR3		Indicates compare/match generation
TIER0–TIER3		Enables IMFA interrupts
TSTR		Enables timer count operation

2.10.3 Description of Operations

Figure 2.51 shows the principle of data output operation using TPC output Group 3. 4-phase non-overlap output is conducted due to H8/3003 hardware and software processing.

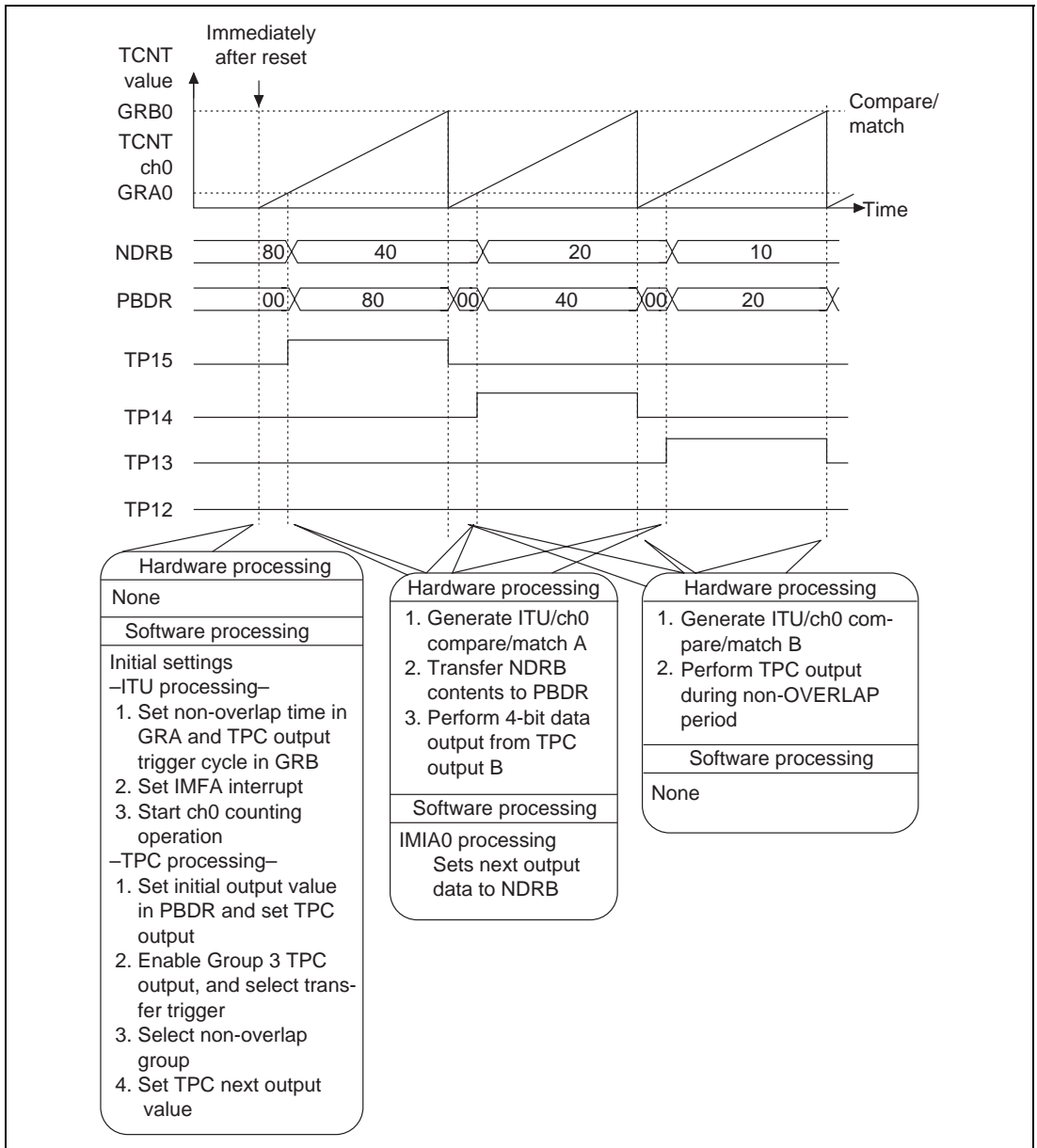


Figure 2.51 Principle of 4-Phase Non-Overlap Output Operation

2.10.4 Description of Software

The software for 4-bit × 4 system output is described below.

Table 2.48 Description of Modules

Module Name	Label Name	Function
Main routine	TPC16MN	Conducts TPC and ITU initial setting
Data setting 0	SETDAT0	Sets next output data in TPC's NDRA (Group 0)
Data setting 1	SETDAT1	Sets next output data in TPC's NDRA (Group 1)
Data setting 2	SETDAT2	Sets next output data in TPC's NDRB (Group 2)
Data setting 3	SETDAT3	Sets next output data in TPC's NDRB (Group 3)

Table 2.49 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
INC_ADR0	Stores data to be added to NDAT_TAB address	1 byte	Main routine	Output
			Data setting 0	Input
INC_ADR1	Stores data to be added to NDAT_TAB address	1 byte	Main routine	Output
			Data setting 1	Input
INC_ADR2	Stores data to be added to NDAT_TAB address	1 byte	Main routine	Output
			Data setting 2	Input
INC_ADR3	Stores data to be added to NDAT_TAB address	1 byte	Main routine	Output
			Data setting 3	Input

Table 2.50 Description of Internal Registers

Register Name	Function	Module Name	
TPC	PADDR	Enables TP7–TP0 TPC output	Main routine
	PBDDR	Enables TP15–TP8 TPC output	Main routine
	PADR	Stores TP7–TP0 output pattern data	Main routine
	PBDR	Stores TP15–TP8 output pattern data	Main routine
	TPMR	Sets TP15–TP0 non-overlap output	Main routine
	TPCR	Sets TP15–TP0 output trigger to ITU ch0 compare/ match	Main routine
	NDERA	Enables TPC outputs TP7–TP0	Main routine
	NDERB	Enables TPC outputs TP15–TP8	Main routine
	NDRA	Sets next output pattern for TP7–TP0	Main routine Data setting 1, 2
	NDRB	Sets next output pattern for TP15–TP8	Main routine Data setting 3, 4
ITU	GRA0–GRA3	Sets non-overlap time	Main routine
	GRB0–GRB3	Sets TPC output trigger cycle	Main routine
	TCR0–TCR3	Sets TCR as follows: <ul style="list-style-type: none"> • Counter clear with GRB compare/match • Count by internal clock ϕ 	Main routine
	TSR0–TSR3	Indicates compare/match generation	Main routine
	TIER0–TIER3	Enables IMFA interrupts	Main routine
	TSTR	Enables TCNT count operation	Main routine

Description of RAM: RAM not used other than for arguments in this sample task.

Table 2.51 Description of Data Tables

Table Name	Function	Data Length	Data Volume
NDAT_TAB0	Stores output data from TPC Group 0	1 byte	5 bytes
NDAT_TAB1	Stores output data from TPC Group 1	1 byte	5 bytes
NDAT_TAB2	Stores output data from TPC Group 2	1 byte	5 bytes
NDAT_TAB3	Stores output data from TPC Group 3	1 byte	5 bytes

Description of General Registers: Used as work registers during data setting in all routines.

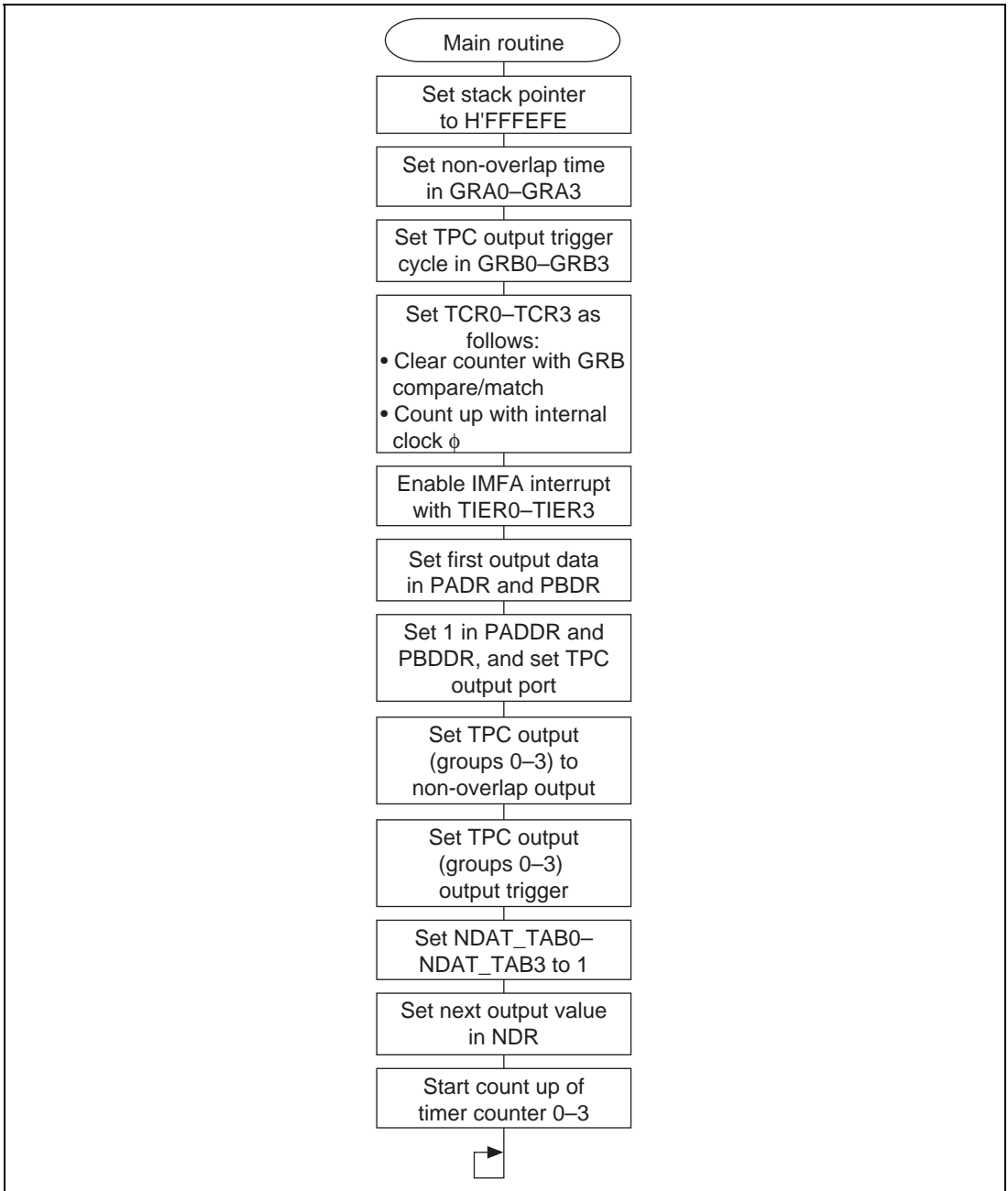


Figure 2.52 Main Routine Flowchart

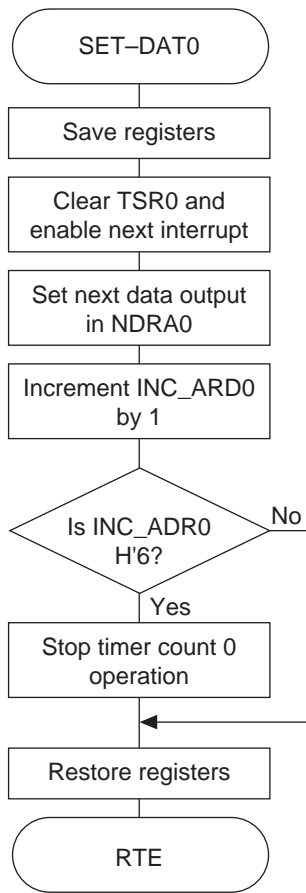


Figure 2.53 SET-DAT0 Flowchart

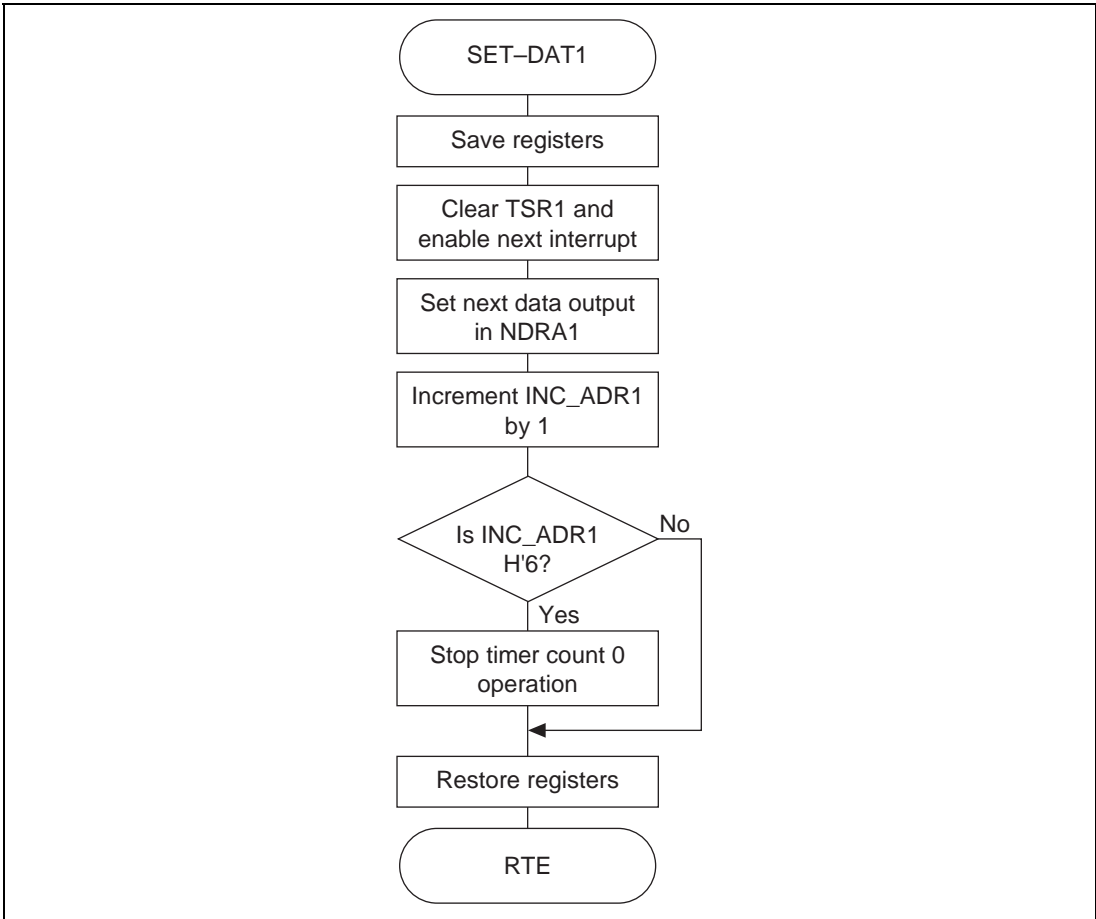


Figure 2.54 SET-DAT1 Flowchart

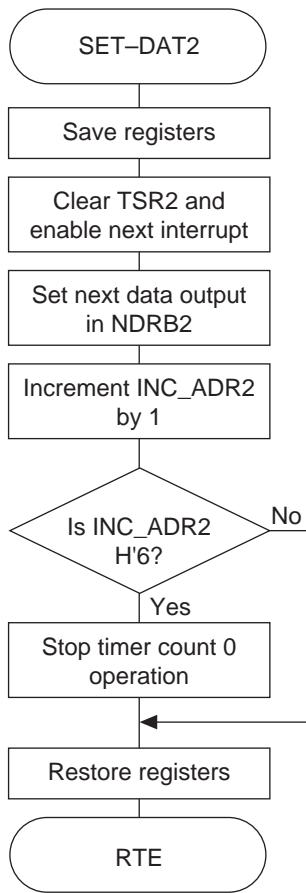


Figure 2.55 SET-DAT2 Flowchart

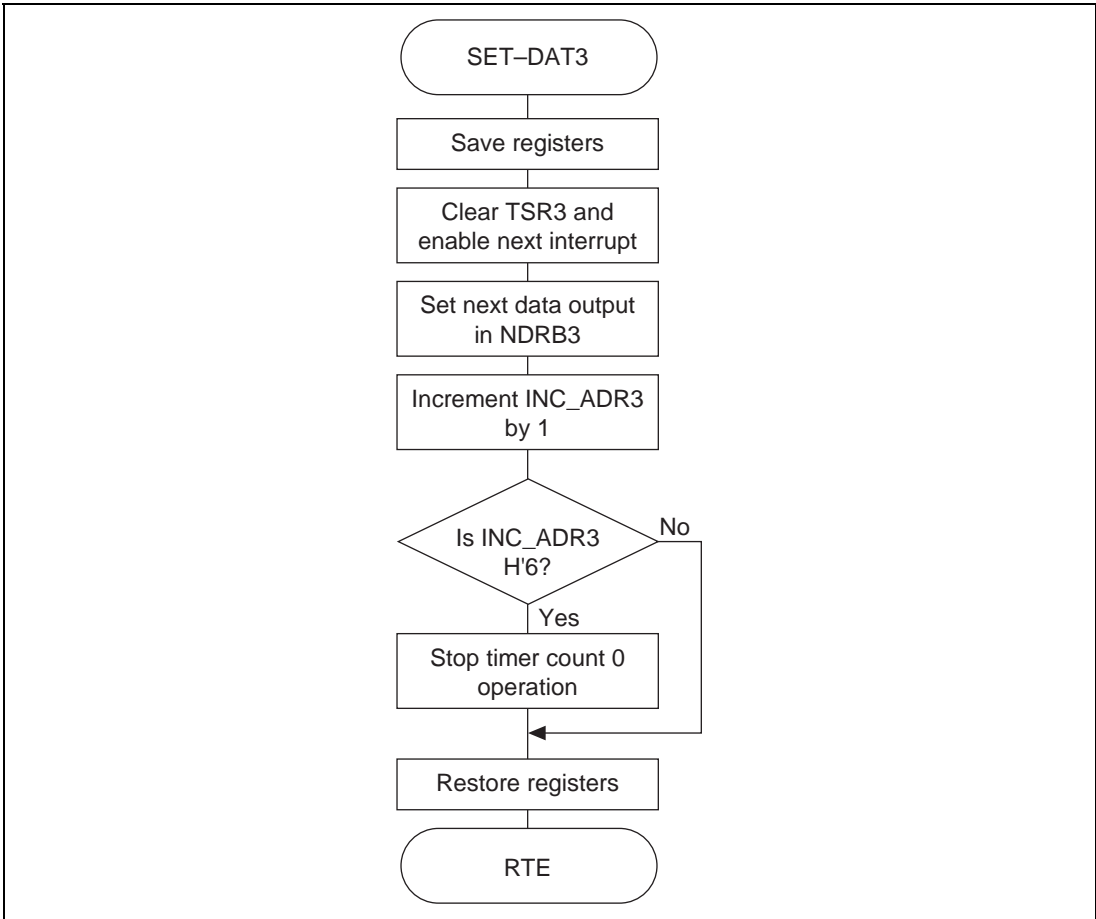


Figure 2.56 SET-DAT3 Flowchart

2.10.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6
7          7          .CPU      300HA
8 000000    8          .SECTION  VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES      .DATA.L   TPC16MN
11         11 ;
12 00001C    12          .ORG      H'00001C
13 00001C 00010000 13 NMI      .DATA.L   TPC16MN
14         14 ;
15 000030    15          .ORG      H'000030
16 000030 00010000 16 IRQ0     .DATA.L   TPC16MN
17 000034 00010000 17 IRQ1     .DATA.L   TPC16MN
18 000038 00010000 18 IRQ2     .DATA.L   TPC16MN
19 00003C 00010000 19 IRQ3     .DATA.L   TPC16MN
20 000040 00010000 20 IRQ4     .DATA.L   TPC16MN
21 000044 00010000 21 IRQ5     .DATA.L   TPC16MN
22 000048 00010000 22 IRQ6     .DATA.L   TPC16MN
23 00004C 00010000 23 IRQ7     .DATA.L   TPC16MN
24 000050 00010000 24 WOVI0    .DATA.L   TPC16MN
25 000054 00010000 25 CMI      .DATA.L   TPC16MN
26         26 ;
27 000060    27          .ORG      H'000060
28 000060 000100CB 28 IMIA0    .DATA.L   SETDAT0
29 000064 00010000 29 IMIB0    .DATA.L   TPC16MN
30 000068 00010000 30 OVI0     .DATA.L   TPC16MN
31         31 ;
32 000070    32          .ORG      H'000070
33 000070 00010102 33 IMIA1    .DATA.L   SETDAT1
34 000074 00010000 34 IMIB1    .DATA.L   TPC16MN
35 000078 00010000 35 OVI1     .DATA.L   TPC16MN
36         36 ;
37 000080    37          .ORG      H'000080
38 000080 0001013C 38 IMIA2    .DATA.L   SETDAT2
39 000084 00010000 39 IMIB2    .DATA.L   TPC16MN
40 000088 00010000 48 OVI2     .DATA.L   TPC16MN
41         41 ;
42 000090    42          .ORG      H'000090
43 000090 00010176 43 IMIA3    .DATA.L   SETDAT3

```



```

44 000094 00010000      44 IMIB3      .DATA.L      TPC16MN
45 000098 00010000      45 OVI3       .DATA.L      TPC16MN
46                                     46 ;
47 0000A0                                     47          .ORG          H'0000A0
48 0000A0 00010000      48 IMMIA4     .DATA.L      TPC16MN
49 0000A4 00010000      49 IMIB4      .DATA.L      TPC16MN
50 0000A8 00010000      50 OVI4       .DATA.L      TPC16MN
51                                     51 ;
52 0000B0                                     52          .ORG          H'0000B0
53 0000B0 00010000      53 DEND0A     .DATA.L      TPC16MN
54 0000B4 00010000      54 DEND0B     .DATA.L      TPC16MN
55 0000B8 00010000      55 DEND1A     .DATA.L      TPC16MN
56 0000BC 00010000      56 DEND1B     .DATA.L      TPC16MN
57 0000C0 00010000      57 DEND2A     .DATA.L      TPC16MN
58 0000C4 00010000      58 DEND2B     .DATA.L      TPC16MN
59 0000CB 00010000      59 DEND3A     .DATA.L      TPC16MN
60 0000CC 00010000      60 DEND3B     .DATA.L      TPC16MN
61 0000D0 00010000      61 ERI0       .DATA.L      TPC16MN
62 0000D4 00010000      62 RXI0       .DATA.L      TPC16MN
63 0000D8 00010000      63 TXI0       .DATA.L      TPC16MN
64 0000DC 00010000      64 TEI0       .DATA.L      TPC16MN
65 0000E0 00010000      65 ERI1       .DATA.L      TPC16MN
66 0000E4 00010000      66 RXI1       .DATA.L      TPC16MN
67 0000E8 00010000      67 TXI1       .DATA.L      TPC16MN
68 0000EC 00010000      68 TEI1       .DATA.L      TPC16MN
69 0000F0 00010000      69 ADI        .DATA.L      TPC16MN
70                                     70 ;
71                                     71 ;*****
72                                     72 ;* *
73                                     73 ;*      RAM ALLOCATION *
74                                     74 ;* *
75                                     75 ;*****
76                                     76 ;
77 FFFF00                                     77          .SECTION      RAM,DATA,LOCATE=H'FFFF00
78                                     78 ;
79 FFFF00 00000001      79 INC_ADR0   .RES.B      1          ;Next data address 0
80 FFFF01 00000001      80 INC_ADR1   .RES.B      1          ;Next data address 1
81 FFFF02 00000001      81 INC_ADR2   .RES.B      1          ;Next data address 2
82 FFFF03 00000001      82 INC_ADR3   .RES.B      1          ;Next data address 3
83                                     83 ;
84                                     84 ;*****
85                                     85 ;* *
86                                     86 ;*      SYMBOL DEFINITIONS *
87                                     87 ;* *
88                                     88 ;*****
89                                     89 ;

```

```

90 ;***** ITU *****
91 00FFFF60 91 TSTR: .EQU H'FFFF60 ;Timer start register
92 00000003 92 STR3: .EQU 3 ;Timer 3 count start bit
93 00000002 93 STR2: .EQU 2 ;Timer 2 count start bit
94 00000001 94 STR1: .EQU 1 ;Timer 1 count start bit
95 00000000 95 STR0: .EQU 0 ;Timer 0 count start bit
96 ;
97 ;***** ITU ch0 *****
98 00FFFF64 98 TCR0: .EQU H'FFFF64 ;timer control register
99 00FFFF66 99 TIER0: .EQU H'FFFF66 ;timer interrupt enable
;register
100 00FFFF67 100 TSR0 .EQU H'FFFF67 ;timer status register
101 00FFFF6A 101 GRA0H: .EQU H'FFFF6A ;General register A
102 00FFFF6C 102 GRB0H: .EQU H'FFFF6C ;General register B
103 ;
104 ;***** ITU ch1 *****
105 00FFFF6E 105 TCR1: .EQU H'FFFF6E ;timer control register
106 00FFFF70 106 TIER1: .EQU H'FFFF70 ;timer interrupt enable
;register
107 00FFFF71 107 TSR1: .EQU H'FFFF71 ;timer status register
108 00FFFF74 108 GRA1H: .EQU H'FFFF74 ;General register A
109 00FFFF76 109 GRB1H: .EQU H'FFFF76 ;General register B
110 ;
111 ;***** ITU ch2 *****
112 00FFFF78 112 TCR2: .EQU H'FFFF78 ;timer control register
113 00FFFF7A 113 TIER2: .EQU H'FFFF7A ;timer interrupt enable
;register
114 00FFFF7B 114 TSR2: .EQU H'FFFF7B ;timer status register
115 00FFFF7E 115 GRA2H: .EQU H'FFFF7E ;General register A
116 00FFFF80 116 GRB2H: .EQU H'FFFF80 ;General register B
117 ;
118 ;***** ITU ch3 *****
119 00FFFF82 119 TCR3: .EQU H'FFFF82 ;timer control register
120 00FFFF84 120 TIER3: .EQU H'FFFF84 ;timer interrupt enable
;register
121 00FFFF85 121 TSR3: .EQU H'FFFF85 ;timer status register
122 00FFFF88 122 GRA3H: .EQU H'FFFF88 ;General register A
123 00FFFF8A 123 GRB3H: .EQU H'FFFF8A ;General register B
124 ;
125 ;***** TPC *****
126 00FFFA0 126 TPMP: .EQU H'FFFA0 ;TPC output mod register
127 00FFFA1 127 TPCR: .EQU H'FFFA1 ;TPC output control register
128 00FFFA2 128 NDERB: .EQU H'FFFA2 ;Next data enable register B
129 00FFFA3 129 NDERA: .EQU H'FFFA3 ;Next data enable register A
130 00FFFA4 130 NDRB3: .EQU H'FFFA4 ;Next data register 3
131 00FFFA5 131 NDRA1: .EQU H'FFFA5 ;Next data register 1

```

```

132      00FFFA6      132 NDRB2:  .EQU      H'FFFA6 ;Next data register 2
133      00FFFA7      133 NDRA0:  .EQU      H'FFFA7 ;Next data register 0
134      00FFFD1      134 PADDR:  .EQU      H'FFFD1 ;Port A data direction
                                ;register
135      00FFFD3      135 PADR:   .EQU      H'FFFD3 ;Port A data register
136      00FFFD4      136 PBDDR:  .EQU      H'FFFD4 ;Port B data direction
                                ;register
137      00FFFD6      137 PBDR:   .EQU      H'FFFD6 ;Port B data register
138      ;
139      ;*****
140      ;*
141      ;*          MAIN PROGRAM : TPC16MN
142      ;*
143      ;*****
144      ;
145 010000      145          .SECTION  PROG, CODE, LOCATE=H'010000
146      ;
147      00010000      147 TPC16MN: .EQU      $
148 010000      7A0700FFFEFE      148          MOV.L     #H'FFFEFE, SP      ;Init. stack pointer
149      ;
150      ;***** INITIALZE ITU ch0 *****
151 010006      7900000A      151          MOV.W     #H'000A, R0
152 01000A      6B80FF6A      152          MOV.W     R0, @GRA0H      ;Set non overlap time
153 01000E      790001F4      153          MOV.W     #D'500, R0
154 010012      6B80FF6C      154          MOV.W     R0, @GRB0H      ;Set timer counter
155 010016      F840      155          MOV.B     #H'40, R0L
156 01001B      3864      156          MOV.B     R0L, @TCR0      ;Init. TCR0(1->CCLR1)
157 01001A      F801      157          MOV.B     #H'01, R0L
158 01001C      3866      158          MOV.B     R0L, @TIER0      ;Init. TIER0(1->IMIEA)
159      ;
160      ;***** INITIALZE ITU ch1 *****
161 01001E      7900000A      161          MOV.W     #H'000A, R0
162 010022      6B80FF74      162          MOV.W     R0, @GRA1H      ;Set non overlap time
163 010026      790003E8      163          MOV.W     #D'1000, R0
164 01002A      6B80FF76      164          MOV.W     R0, @GRB1H      ;Set timer counter
165 01002E      F840      165          MOV.B     #H'40, R0L
166 010030      386E      166          MOV.B     R0L, @TCR1      ;Init. TCR1(1->CCLR1)
167 010032      F801      167          MOV.B     #H'01, R0L
168 010034      3870      168          MOV.B     R0L, @TIER1      ;Init. TIER1(1->IMIEA)
169      ;
170      ;***** INITIALZE ITU ch2 *****
171 010036      7900000A      171          MOV.W     #H'000A, R0
172 01003A      6B80FF7E      172          MOV.W     R0, @GRA2H      ;Set non overlap time
173 01003E      790005DC      173          MOV.W     #D'1500, R0
174 010042      6B80FF80      174          MOV.W     R0, @GRB2H      ;Set timer counter
175 010046      F840      175          MOV.B     #H, 40, R0L

```

```

176 010048 3878          176          MOV.B      R0L,@TCR2          ;Init. TCR2(1->CCLR1)
177 01004A F801          177          MOV.B      #H'01,R0L
178 01004C 387A          178          MOV.B      R0L,@TIER2          ;Init. TIER2(1->IMIEA)
179
180                      ;***** INITIALZE ITU ch3 *****
181 01004E 7900000A      181          MOV.W      #H'000A,R0
182 010052 6B80FF88      182          MOV.W      R0,@GRA3H          ;Set non overlap time
183 010056 790007D0      183          MOV.W      #D'2000,R0
184 01005A 6B80FFBA      184          MOV.W      R0,@GRB3H          ;Set timer counter
185 01005E F840          185          MOV.B      #H'40,R0L
186 010060 3882          186          MOV.B      R0L,@TCR3          ;Init. TCR3(1->CCLR1)
187 010062 F801          187          MOV.B      #H'01,R0L
188 010064 3884          188          MOV.B      R0L,@TIER3          ;Init. TIER3(1->IMIEA)
189
190                      ;***** INITIALZE TPC *****
191 010066 6A28000101AC    191          MOV.B      @NDAT_TAB0,R0L
192 01006C 6A20000101B1    192          MOV.B      @NDAT_TAB1,R0H
193 010072 1480          193          OR.B       R0L,R0H
194 010074 30D3          194          MOV.B      R0H,@PADR          ;Set first output data
                                   ;(Gr1,Gr0)
195 010076 6A28000101B6    195          MOV.B      @NDAT_TAB2,R0L
196 01007C 6A20000101BB    196          MOV.B      @NDAT_TAB3,R0H
197 010082 1480          197          OR.B       R0L,R0H
198 010084 30D6          198          MOV.B      R0H,@PBDR          ;Set first outupt data
                                   ;(Gr3,Gr2)
199 010086 F8FF          199          MOV.B      #H'FF,R0L
200 010088 38D1          200          MOV.B      R0L,@PADDR
201 0100BA 38D4          201          MOV.B      R0L,@PBDDR
202 0100BC 38A3          202          MOV.B      R0L,@NDERA
203 01008E 38A2          203          MOV.B      R0L,@NDERB          ;Set output TPC
204 010090 38A0          204          MOV.B      R0L,@TPMR          ;Set output non
                                   ;overlap
205 010092 F8E4          205          MOV.B      #H'E4,R0L
206 010094 38A1          206          MOV.B      R0L,@TPCR          ;Init. TPCR
207 010096 6A28000101AD    207          MOV.B      @NDAT_TAB0+1,R0L
208 01009C 38A7          208          MOV.B      R0L,@NDRA0          ;Set second output
                                   ;data(Gr0)
209 01009E 6A28000101B2    209          MOV.B      @NDAT_TAB1+1,R0L
210 0100A4 38A5          210          MOV.B      R0L,@NDRA1          ;Set second output
                                   ;data (Gr1)
211 0100A6 6A28000101B7    211          MOV.B      @NDAT_TAB2+1,R0L
212 0100AC 38A6          212          MOV.B      R0L,@NDRB2          ;Set second output
                                   ;data(Gr2)
213 0100AE 6A28000101BC    213          MOV.B      @NDAT_TAB3+1,R0L
214 0100B4 38A4          214          MOV.B      R0L,@NDRB3          ;Set second output
                                   ;data(Gr3)

```

```

215                                     215 ;
216 0100B6 F802                         216         MOV.B         #H'02,R0L
217 0100B8 3800                         217         MOV.B         R0L,@INC_ADR0 ;Init. INC_ADR0
218 0100BA 3801                         218         MOV.B         R0L,@INC_ADR1 ;Init. INC_ADR1
219 0100BC 3802                         219         MOV.B         R0L,@INC_ADR2 ;Init. INC_ADR2
220 0100BE 3803                         220         MOV.B         R0L,@INC_ADR3 ;Init. INC_ADR3
221 0100C0 F8EF                         221         MOV.B         #H'EF,R0L
222 0100C2 3860                         222         MOV.B         R0L,@TSTR ;Start ITU ch0-3
223 0100C4 0700                         223         LDC.B         #0,CCR ;Enable interrupt
224                                     224 ;
225 0100C6 40FE                         225 TPC16MN99 BRA         TPC16MN99
226                                     226 ;
227                                     227 ;*****
228                                     228 ;* *
229                                     229 ;* NAME : SINDAT0 (Set next data to TPC Gr0) *
230                                     230 ;* *
231                                     231 ;*****
232                                     232 ;* *
233                                     233 ;* ENTRY : INC_ADR0 (Address of output data) *
234                                     234 ;* RETURNS : NOTHING *
235                                     235 ;* *
236                                     236 ;*****
237                                     237 ;
238 0100C8 01006DF0                     238 SETDAT0: PUSH.L     ER0 ;Escape register
239 0100CC 01006DF1                     239         PUSH.L     ER1
240 0100D0 F800                         240         MOV.B         #0,R0L ;TSR clr
241 0100D2 2867                         241         MOV.B         #TSR0,R0H
242 0100D4 3867                         242         MOV.B         R0L,@TSR0
243 0100D6 7A0000000000                 243         MOV.L         #D'0,ER0 ;Register clr
244 0100DC 2800                         244         MOV.B         @INC_ADR0,R0L
245 0100DE 7A01000101AC                 245         MOV.L         #NDAT_TAB0,ER1
246 0100E4 0A81                         246         ADD.L         ER0,ER1
247 0100E6 6810                         247         MOV.B         @ER1,R0H
248 0100E8 30A7                         248         MOV.B         R0H,@NDRA0 ;Set NDRA0(Gr0)
249 0100EA 8801                         249         ADD.B         #D'1,R0L ;NDAT_TAB0 Address
                                         ;incliment
250 0100EC 3800                         250         MOV.B         R0L,@INC_ADR0 ;INC_ADR0+1 ->INCADR0
251 0100EE A805                         251         CMP.B         #D'5,R0L ;All output end?
252 0100F0 58600004                     252         BNE         SETDAT099 ;No
253 0100F4 7F607200                     253         BCLR.B     #STR0,@TSTR ;ITU ch0 Stop
254 0100F8 01006D71                     254 SETDAT099: POP.L     ER1
255 0100FC 01006D70                     255         POP.L     ER0
256 010100 5670                         256         RTE
257                                     257 ;

```

```

258 ;*****
259 ;*
260 ;* NAME : SINDAT1 (Set next data to TPC Gr1) *
261 ;*
262 ;*****
263 ;*
264 ;* ENTRY : INC_ADR1 (Address of outupt data) *
265 ;* RETURNS : NOTHING *
266 ;*
267 ;*****
268 ;
269 010102 01006DF0 269 SETDAT1: PUSH.L ER0 ;Escape register
270 010106 01006DF1 270 PUSH.L ER1
271 01010A F800 271 MOV.B #0,R0L ;TSR clr
272 01010C 2071 272 MOV.B @TSR1,R0H
273 01010E 3871 273 MOV.B R0L,@TSR1
274 010110 7A0000000000 274 MOV.L #D'0,ER0
275 010116 2801 275 MOV.B @INC_ADR1,R0L
276 010118 7A01000101B1 276 MOV.L #NDAT_TAB1,ER1
277 01011E 0A81 277 ADD.L ER0,ER1
278 010120 6810 278 MOV.B @ER1,R0H
279 010122 30A5 279 MOV.B R0H,@NDRA1 ;Set NDRA1(Gr1)
280 010124 8801 280 ADD.B #D'1,R0L ;NDAT_TAB1 Address
;incliment
281 010126 3801 281 MOV.B R0L,@INC-ADR1 ;INC_ADR1+1 ->
;INC_ADR1
282 010128 A805 282 CMP.B #D'5,R0L ;All output end?
283 01012A 58600004 283 BNE SETDAT199 ;No
284 01012E 7F607210 284 BCLR.B #STR1,@TSTR ;ITU ch1 Stop
285 010132 01006D71 285 SETDAT199: POP.L ER1
286 010136 01006D70 286 POP.L ER0
287 01013A 5670 287 RET
288 ;
289 ;*****
290 ;*
291 ;* NAME : SINDAT2 (Set next data to TPC Gr2) *
292 ;*
293 ;*****
294 ;*
295 ;* ENTRY : INC_ADR2 (Address Of output data) *
296 ;* RETURNS : NOTHING *
297 ;*
298 ;*****
299 ;
300 01013C 01006DF0 300 SETDAT2: PUSH.L ER0 ;Escape register
301 010140 01006DF1 301 PUSH.L ER1

```

```

302 010144 F800          302          MOV.B      #0,R0L          ;TSR clr
303 010146 207B         303          MOV.B      @TSR2,R0H
304 010148 387B         304          MOV.B      R0L,@TSR2
305 01014A 7A0000000000 305          MOV.L      #D'0,ER0
306 010150 2802         306          MOV.B      @INC_ADR2,R0L
307 010152 7A01000101B6 307          MOV.L      #NDAT_TAB2,ER1
308 010158 0A81         308          ADD.L      ER0,ER1
309 01015A 6810         309          MOV.B      @ER1,R0H
310 01015C 30A6         310          MOV.B      R0H,@NDRB2      ;Set NDRB2(Gr2)
311 01015E 8801         311          ADD.B      #D'1,R0L          ;NDAT_TAB2 Address
                                   ;incliment

312 010160 3802         312          MOV.B      R0L,@INC_ADR2;INC_ADR2+1 -> INC_ADR2
313 010162 A805         313          CMP.B      #D'5,R0L          ;All output end?
314 010164 58600004       314          BNE        SETDAT299      ;No
315 010168 7F607220       315          BCLR.B     #STR2,@TSTR     ;ITU ch2 Stop
316 01016C 01006D71       316 SETDAT299: POP.L      ER1
317 010170 01006D70       317          POP.L      ER0
318 010174 5670         318          RTE
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
331 010176 01006DF0       331 SETDAT3: PUSH.L      ER0          ;Escape register
332 01017A 01006DF1       332          PUSH.L      ER1
333 01017E F800          333          MOV.B      #0,R0L          ;TSR clr
334 010180 2085         334          MOV.B      @TSR3,R0H
335 010182 3885         335          MOV.B      R0L,@TSR3
336 010184 7A0000000000 336          MOV.L      #D'0,ER0
337 01018A 2803         337          MOV.B      @INC_ADR3,R0L
338 01018C 7A01000101BB 338          MOV.L      #NDAT_TAB3,ER1
339 010192 0A81         339          ADD.L      ER0,ER1
340 010194 6810         340          MOV.B      @ER1,R0H
341 010196 30A4         341          MOV.B      R0H,@NDRB3      ;Set NDRB3(Gr3)
342 010198 8801         342          ADD.B      #D'1,R0L          ;NDAT_TAB3 Address
                                   ;incliment

343 01019A 3803         343          MOV.B      R0L,@INC_ADR3      ;INCADR3+1 -> INC_ADR3
344 01019C A805         344          CMP.B      #D'5,R0L          ;All output end?
345 01019E 58600004       345          BNE        SETDAT399      ;No

```

```

346 0101A2 7F607230      346          BCLR.B      #STR3,@TSTR      ;ITU ch3 Stop
347 0101A6              347 SETDAT399: POP.L      ER1
348 0101A6 01006D70      348          POP.L      ERO
349 0101AA 5670          349          RTE
350                      350 ;
351                      351 ;*****
352                      352 ;
353                      353 ;*          DATA TABLE (OUTPUT PATTERN)
354                      354 ;*
355                      355 ;*****
356 0101AC 0908070605      356 NDAT_TAB0 .DATA.B      H'09,H'08,H'07,H'06,H'05
357 0101B1 8070605040      357 NDAT_TAB1 .DATA.B      H'80,H'70,H'60,H'50,H'40
358 0101B6 0706050403      358 NDAT_TAB2 .DATA.B      H'07,H'06,H'05,H'04,H'03
359 0101BB 6050403020      359 NDAT_TAB3 .DATA.B      H'60,H'50,H'40,H'30,H'20
360                      360 ;
361                      361          .END

****TOTAL ERRORS      0
****TOTAL WARNINGS    0

```



2.11 Asynchronous SCI

MCU: H8/3003

Function: SCI (asynchronous mode)

2.11.1 Specifications

- Transmit/receive of 3-byte data in asynchronous mode between H8/3003 and console (figure 2.57)
- Transmission format: 9600 bps, 8-bit data, 1 stop bit and non-parity bit
- Communication control by RTS and CTS

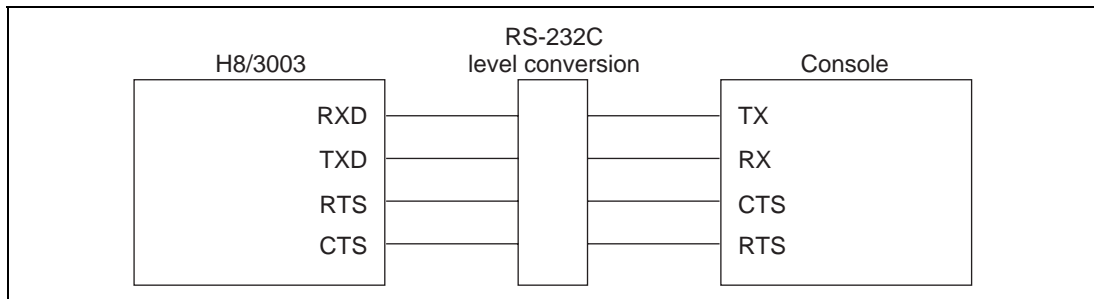


Figure 2.57 Block Diagram of Asynchronous SCI with H8/3003

2.11.2 Description of Functions Used

In this sample task, SCI/ch0 is used for data transmit and receive. PORT 1 is used for communication control (RTS and CTS).

Figure 2.58 shows a block diagram of SCI transmit block used in the sample task. In this sample task, the following SCI functions are used in transmission of data to the console:

- Asynchronous mode data communication synchronized in character units (asynchronous mode)
- Interrupt generation at end of transmit (TEI interrupt)

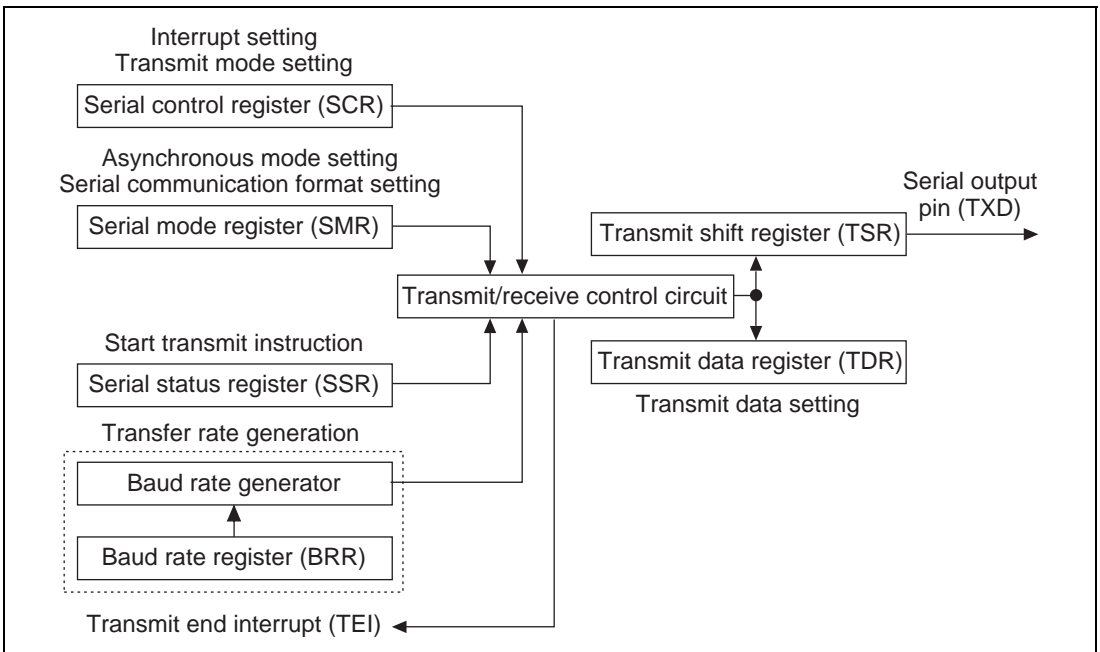


Figure 2.58 Block Diagram of SCI Transmit

Figure 2.59 shows a block diagram of the SCI receive block used in this sample task. The following SCI functions are used for receiving data from the console:

- Asynchronous mode data communication synchronized in character units (asynchronous mode)
- Interrupt generation function at end of receive (RXI interrupt)

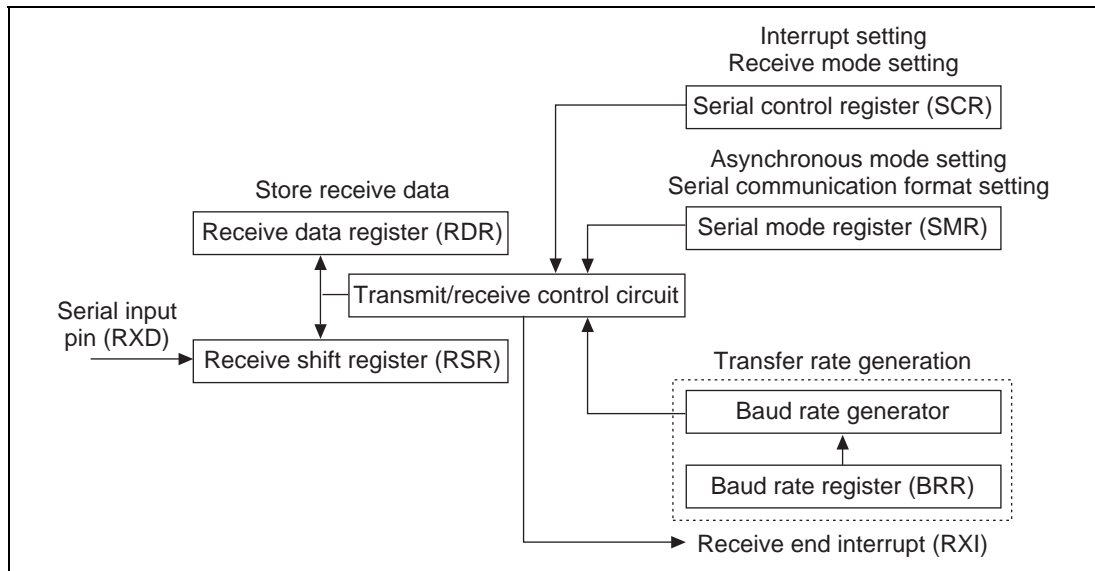


Figure 2.59 SCI Receive Block Diagram

Table 2.52 shows the function allocation in this sample task. SCI functions are allocated for interfacing with the console.

Table 2.52 SCI Function Allocation

SCI Function	Function
RXD	Receive data from console
TXD	Transmit data to console
SMR	Set SCI asynchronous mode and transfer format
SCR	Enable transmit/receive interrupts and set SCI in transmit/receive mode
SSR	Start transmission with TDRE
RDR	Store data receive from console
TDR	Set data for transmission to console
BRR	Set the transfer rate

2.11.3 Description of Operations

Figure 2.60 shows the principle of operation. Interface with the console is conducted by hardware and software processing with the timing shown in figure 2.60.

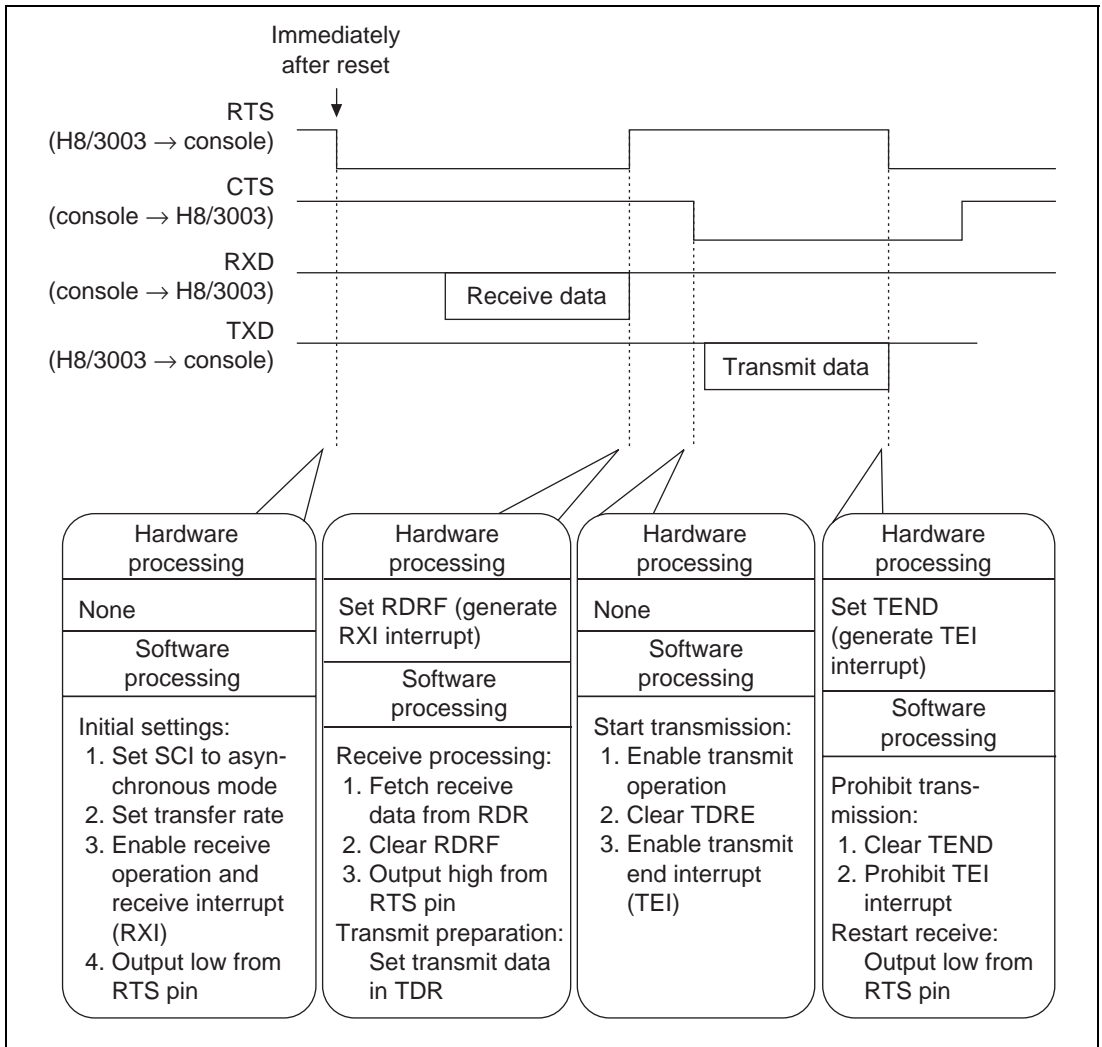


Figure 2.60 Principle of Asynchronous SCI Operation

2.11.4 Description of Software

The software for asynchronous SCI is described below.

Table 2.53 Description of Modules

Module Name	Label Name	Function
Main routine	ASCMN	SCI initial setting and transmit/receive management
Data receive end	ASCRX	Activate RXI interrupt and receive data
Data transmit end	ASCTE	Activate TEI interrupt to notify of end of transmission

Table 2.54 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
R6L	Set data received from console	1 byte	Data receive end	O
			Main routine	I
RXENDF	Flag indicated end of receive 1: receive end, 0: receiving	1 bit	Data receive end	O
			Main routine	I
TXENDF	Flag indicating end of transmit 1: transmit end, 0: transmitting	1 bit	Data transmit end	O
			Main routine	I

Table 2.55 Description of Internal Registers

Register Name	Function	Module Name
SMR	Sets SCI mode (asynchronous mode), transfer format and clock select to baud rate generator (f clock input)	Main routine
SCR	Sets to enable interrupts (RXI, TEI) and enable SCI transmit/receive	Main routine
SSR	Specifies start of transmission by clearing TDRE (b7)	Main routine
RDR	Sets data received from console	Data receive end
TDR	Sets data for transmission to console	Main routine
BRR	Sets transfer rate	Main routine
P9DDR	Sets PORT 9 I/O	Main routine
P9DR	Conducts PRQ pin and CTS pin operations	Main routine

Table 2.56 Description of General Registers

Module Name	Register Name	Function
Main routine	R0L	Used as work register during setting of SCI and argument RAM
	R1L	Used as counter for number of transmit/receive data
	R6L	Used as argument for receive data acceptance
Data receive end	R6L	Used as argument for receive data transfer

Description of RAM: RAM not used other than for arguments in this sample task.

2.11.5 Flowcharts

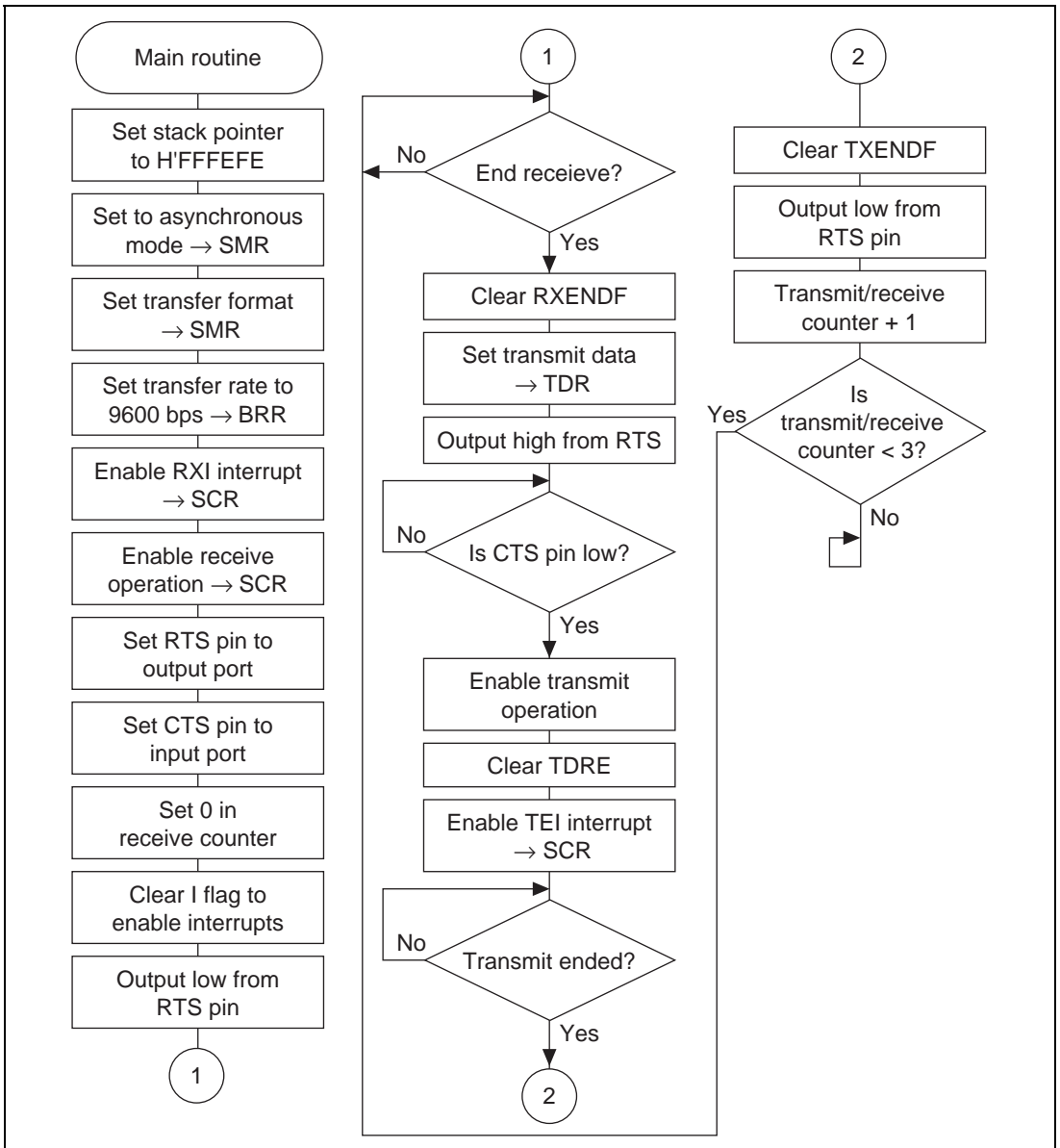


Figure 2.61 Main Routine Flowchart

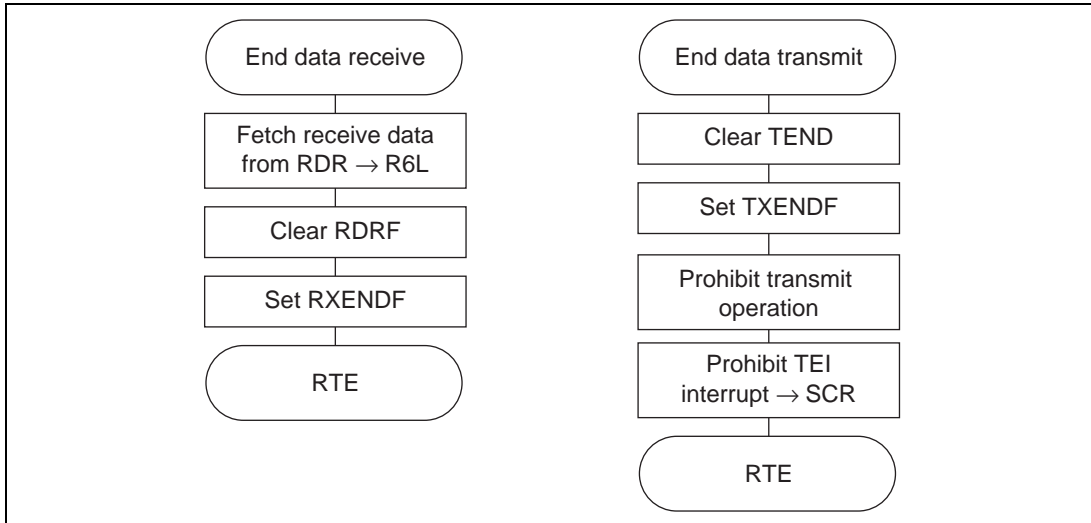


Figure 2.62 End Data Receive and End Data Transmit Flowchart

2.11.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6
7          7          .CPU          300HA
8 000000   8          .SECTION    VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L    ASCMN
11         11 ;
12 00001C   12          .ORG          H'00001C
13 00001C 00010000 13 NMI          .DATA.L    ASCMN
14 000020 00010000 14 TRAP0        .DATA.L    ASCMN
15 000024 00010000 15 TRAP1        .DATA.L    ASCMN
16 000028 00010000 16 TRAP2        .DATA.L    ASCMN
17 00002C 00010000 17 TRAP3        .DATA.L    ASCMN
18         18 ;
19 000030   19          .ORG          H'000030
20 000030 00010000 20 IRQ0        .DATA.L    ASCMN
21 000034 00010000 21 IRQ1        .DATA.L    ASCMN
22 000038 00010000 22 IRQ2        .DATA.L    ASCMN
23 00003C 00010000 23 IRQ3        .DATA.L    ASCMN
24 000040 00010000 24 IRQ4        .DATA.L    ASCMN
25 000044 00010000 25 IRQ5        .DATA.L    ASCMN
26 000048 00010000 26 IRQ6        .DATA.L    ASCMN
27 00004C 00010000 27 IRQ7        .DATA.L    ASCMN
28 000050 00010000 28 WOVI        .DATA.L    ASCMN
29 000054 00010000 29 CMI          .DATA.L    ASCMN
30         30 ;
31 000060   31          .ORG          H'000060
32 000060 00010000 32 IMIA0       .DATA.L    ASCMN
33 000064 00010000 33 IMIB0       .DATA.L    ASCMN
34 000068 00010000 34 OVI0       .DATA.L    ASCMN
35         35 ;
36 000070   36          .ORG          H'000070
37 000070 00010000 37 IMIA1       .DATA.L    ASCMN
38 000074 00010000 38 IMIB1       .DATA.L    ASCMN
39 000078 00010000 39 OVI1       .DATA.L    ASCMN
40         40 ;
41 000080   41          .ORG          H'000080
42 000080 00010000 42 IMIA2       .DATA.L    ASCMN
43 000084 00010000 43 IMIB2       .DATA.L    ASCMN

```

```

44 000088 00010000      44 OVI2      .DATA.L  ASCMN
45                               45 ;
46 000090                               46          .ORG      H'000090
47 000090 00010000      47 IMIA3     .DATA.L  ASCMN
48 000094 00010000      48 IMIB3     .DATA.L  ASCMN
49 000098 00010000      49 OVI3      .DATA.L  ASCMN
50                               50 ;
51 0000A0                               51          .ORG      H'0000A0
52 0000A0 00010000      52 IMIA4     .DATA.L  ASCMN
53 0000A4 00010000      53 IMIB4     .DATA.L  ASCMN
54 0000A8 00010000      54 OVI4      .DATA.L  ASCMN
55                               55 ;
56 0000B0                               56          .ORG      H'0000B0
57 0000B0 00010000      57 DEND0A    .DATA.L  ASCMN
58 0000B4 00010000      58 DEND0B    .DATA.L  ASCMN
59 0000B8 00010000      59 DEND1A    .DATA.L  ASCMN
60 0000BC 00010000      60 DEND1B    .DATA.L  ASCMN
61 0000C0 00010000      61 DEND2A    .DATA.L  ASCMN
62 0000C4 00010000      62 DEND2B    .DATA.L  ASCMN
63 0000C8 00010000      63 DEND3A    .DATA.L  ASCMN
64 0000CC 00010000      64 DEND3B    .DATA.L  ASCMN
65 0000D0 00010000      65 ERI0      .DATA.L  ASCMN
66 0000D4 0001005A      66 RXI0      .DATA.L  ASCIRX
67 0000D8 00010000      67 TXI0      .DATA.L  ASCMN
68 0000DC 00010066      68 TEI0      .DATA.L  ASCITE
69 0000E0 00010000      69 ERI1      .DATA.L  ASCMN
70 0000E4 00010000      70 RXI1      .DATA.L  ASCMN
71 0000E8 00010000      71 TXI1      .DATA.L  ASCMN
72 0000EC 00010000      72 TEI1      .DATA.L  ASCMN
73 0000F0 00010000      73 ADI       .DATA.L  ASCMN
74                               74 ;
75                               75 ;*****
76                               76 ;*
77                               77 ;*      RAM ALLOCATION
78                               78 ;*
79                               79 ;*****
80                               80 ;
81 FFFF00                               81          .SECTION  RAM,DATA,LOCATE=H'FFFF00
82                               82 ;
83 FFFF00 00000001      83 FLAG     .RES.B   1          ;Flag area
84          00000000      84 RXENDF   .EQU    0          ;Receive end flag
85          00000001      85 TXENDF   .EQU    1          ;Transmit end flag
86                               86 ;
87                               87 ;*****
88                               88 ;*
89                               89 ;*      SYMBOL DEFINITIONS

```

```

90          90 ;*
91          91 ;*****
92          92 ;
93          93 ;***** SCI ch0 *****
94 00FFFFB0 94 SMR0 .EQU H'FFFFB0 ;Serial mode register
95          95 ;
96 00FFFFB1 96 BRR0 .EQU H'FFFFB1 ;Bit rate register
97          97 ;
98 00FFFFB2 98 SCR0 .EQU H'FFFFB2 ;Serial control register
99 00000006 99 RIE .EQU 6 ;Receive interrupt enable
100 00000005 100 TE .EQU 5 ;Transmit enable
101 00000004 101 RE .EQU 4 ;Receive enable
102 00000003 102 MPIE .EQU 3 ;Multi processor interrupt
;enable
103 00000002 103 TEIE .EQU 2 ;Transmit end interrupt
;enable
104          104 ;
105 00FFFFB3 105 TDR0 .EQU H'FFFFB3 ;Transmit data register
106          106 ;
107 00FFFFB4 107 SSR0 .EQU H'FFFFB4 ;Serial status register
108 00000007 108 TDRE .EQU 7 ;Transmit data register
;empty
109 00000006 109 RDRF .EQU 6 ;Receive data register full
110          110 ;
111 00FFFFB5 111 RDR0 .EQU H'FFFFB5 ;Receive data register
112 ;***** PORT 9 *****
113 00FFFFD0 113 P9DDR .EQU H'FFFFD0 ;Port 9 data direction
;register
114          114 ;
115 00FFFFD2 115 P9DR .EQU H'FFFFD2 ;Port 9 data register
116 00000004 116 RTS .EQU 4 ;RTS bit
117 00000005 117 CTS .EQU 5 ;CTS bit
118          118 ;
119 ;*****
120          120 ;*
121          121 ;* MAIN PROGRAM : ASCMN *
122          122 ;*
123 ;*****
124          124 ;
125 010000 125 .SECTION PROG, CODE, LOCATE=H'010000
126          126 ;
127 00010000 127 ASCMN: .EQU $
128 010000 7A0700FFFEFE 128 MOV.L #H'FFFEFE, SP ;Set stack pointer
129 010006 F800 129 MOV.B #B'00000000, R0L ;Asynco mode, 8bit, np,
;stop1
130 010008 38B0 130 MOV.B R0L, @SMR0 ;

```

131	01000A	F833	131	MOV.B	#51,R0L	;CKS0,l=0,N=51
132	01000C	38B1	132	MOV.B	R0L,@BRR0	;Then 9600bps
133	01000E	7FB27060	133	BSET.B	#RIE,@SCR0	;Enable RXI interrupt
134	010012	7FB27040	134	BSET.B	#RE,@SCR0	;Enable receive
135	010016	P8D0	135	MOV.B	#H'D0,R0L	;CTS is input port
136	010018	38D0	136	MOV.B	R0L,@P9DDR	;RTS is output port
137			137	;		
138	01001A	F900	138	MOV.B	#H'0,R1L	;Set TXRXCNT=0
139			139	;		
140	01001C	067F	140	ANDC	#B'01111111,CCR	;Clear I-FLAG
141	01001E	7FD27240	141	BCLR.B	#RTS,@P9DR	;RTS is 'Low'
142	010022		142	ASCMN01:		
143	010022	7E007300	143	BTST.B	#RXENDF,@FLAG	;Receive complete ?
144	010026	47FA	144	BEQ	ASCMN01	;No
145			145	;		
146	010028	7F007200	146	BCLR.B	#RXENDF,@FLAG	;Clear RXENDF
147	01002C	3EB3	147	MOV.B	RGL,@TDR0	;Set transmit data ;(receive data)
148	01002E	7FD27040	148	BSET.B	#RTS.@P9DR	;RTS is 'High'
149			149	;		
150	010032		150	ASCMN02:		
151	010032	7ED27350	151	BTST.B	#CTS,@P9DR	;CTS is 'High' ?
152	010036	47FA	152	BEQ	ASCMN02	;No
153			153	;		
154	010038	7FB27050	154	BSET.B	#TE,@SCR0	;Enable transmit
155	01003C	7FB47270	155	BCLR.B	#TDRE,@SSR0	;Start transmit
156	010040	7FB27020	156	BSET.B	@TEIE,@SCR0	;Enable TEI interrupt
157			157	;		
158	010044		158	ASCMN03:		
159	010044	7E007310	159	BTST.B	#TXENDF,@FLAG	;Transmit complete ?
160	010048	47FA	160	BEQ	ASCMN03	;No
161			161	;		
162	01004A	7F007210	162	BCLR.B	#TXENDF,@FLAG	;Clear TXENDF
163	01004E	7FD27240	163	BCLR.B	#RTS,@P9DR	;RTS is 'Low'
164			164	;		
165	010052	0A09	165	INC.B	R1L	;TXRXCNT + 1
166	010054	A903	166	CMP.B	#3,R1L	;TXRXCNT < 3
167	010056	45CA	167	BCS	ASCMN01	;Yes
168			168	;		
169	010058		169	ASCMN99:		
170	010058	40FE	170	BRA	ASCMN99	
171			171	;		

```

172 ;*****
173 ;*
174 ;*      NAME : ASCIRX(SCI receive interrupt) *
175 ;*
176 ;*****
177 ;*
178 ;*      ENTRY : NOTHING *
179 ;*      RETURNS : RXENDF (Receive end flag) *
180 ;*
181 ;*****
182 ;
183      0001005A      183 ASCIRX:  .EQU      $
184 01005A  2EB5      184      MOV.B      @PDR0,R6L      ;Get receive data to R6L
185 01005C  7FB47260  185      BCLR.B      #RDRF,@SSR0      ;Clear RDRF
186 010060  7F007000  186      BSET.B      #RXENDF,@FLAG ;Set RXENDF
187 010064  5670      187      RTE
188
189 ;*****
190 ;*
191 ;*      NAME : ASCITE(SCI transmit interrupt) *
192 ;*
193 ;*****
194 ;*
195 ;*      ENTRY      : NOTHING *
196 ;*      RETURNS    : TXENDF (Transmit end flag) *
197 ;*
198 ;*****
199 ;
200      00010066      200 ASCITE:  .EQU      $
201 010066  7F007010  201      BSET.B      #TXENDF,@FLAG ;Set TXENDF
202 01006A  7F827250  202      BCLR.B      #TE,@SCR0      ;Disable transmit
203 01006E  7F827220  203      BCLR.B      #TEIE,@SCR0     ;Disable TEI interrupt
204 010072  5670      204      RTE
205
206      206      .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0

```

2.12 Simultaneous Transmit/Receive Operation

MCU: H8/3003

Function: SCI (simultaneous transmit/receive)

2.12.1 Specifications

- Receive/transmit of 1 byte data between H8/3003 and H4019 (see figure 2.63).
- Data transmit/receive is in clock sync format. The master side H8/3003's clock is shared with the slave side H4019.
- The H8/3003 transmits data simultaneously with receiving data from the H4019.

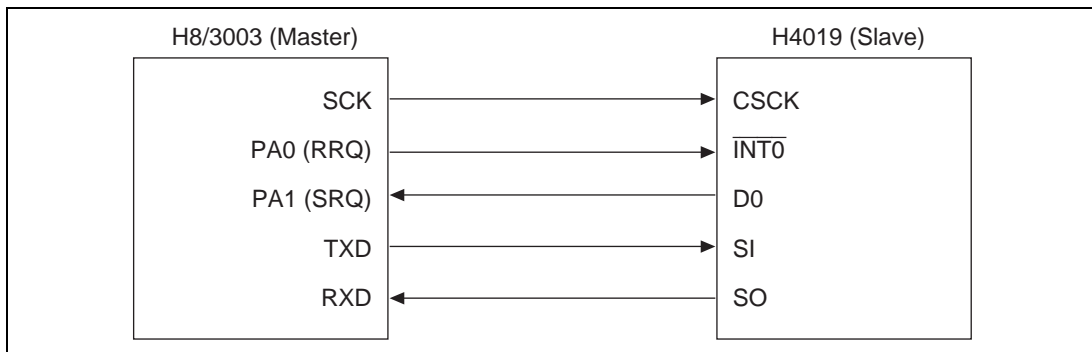


Figure 2.63 Clock Synchronous SCI Interface Due to H8/3003

2.12.2 Description of Functions Used

This sample task conducts data transmit/receive via the SCI. PORT A is used for communication control (RRQ, SRQ).

Figure 2.64 shows a block diagram of the SCI used in this sample task. The following SCI functions are used in simultaneous transmit/receive operations:

- Clock synchronized serial data communication function (clock sync mode)
- Simultaneous transmit and receive function (simultaneous transmit/receive operation)
- Generation of interrupt at end of receive (RXI interrupt)

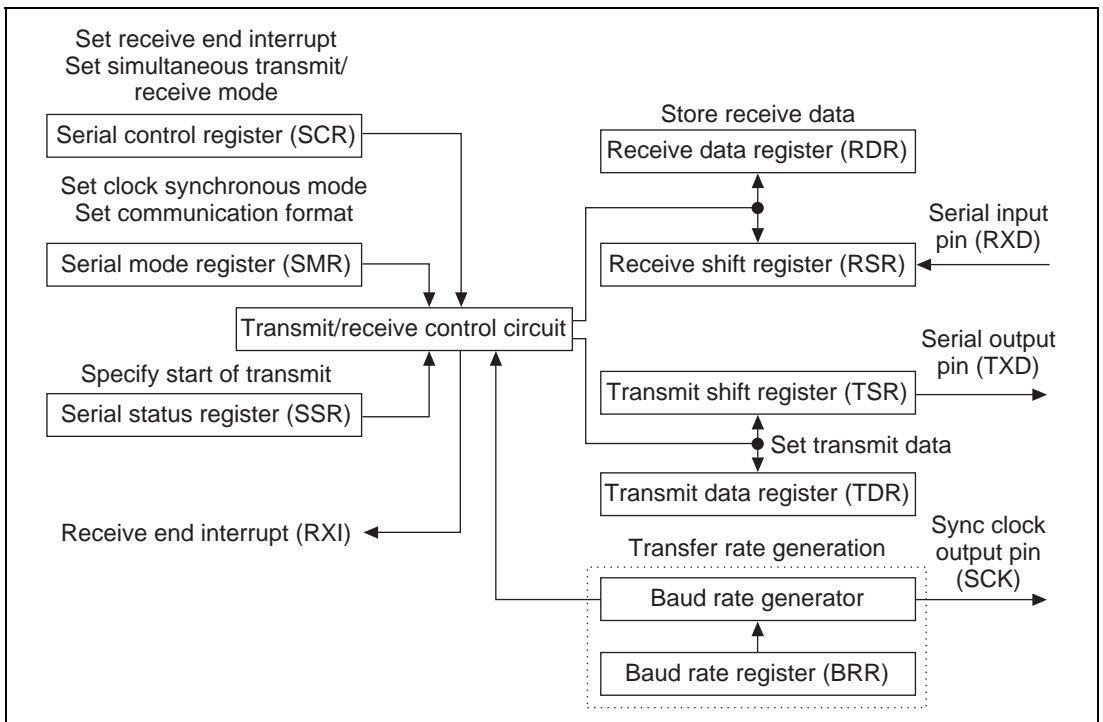


Figure 2.64 SCI Block Diagram

Table 2.57 shows the function allocations for this sample task. SCI functions are allocated for simultaneous data exchange with the H4019.

Table 2.57 SCI Function Allocation

SCI Function	Function
SCK	Transmits sync clock
RXD	Receives data from H4019
TXD	Transmits data to H4019
SMR	Sets SCI to clock sync mode
SCR	Enables receive interrupts and sets SCI to transmit/receive mode
SSR	Starts transmission due to TDRE bit
RDR	Sets data received from H4019
TDR	Sets data for transmission to H4019
BRR	Sets transfer rate
PADDR	Sets PORT A I/O
PADR	Conducts RRQ transmit and SRQ receive

2.12.3 Description of Operations

Figure 2.65 shows the principle of operation. Interfacing with the H4019 is conducted due to hardware and software processing with the timing indicated in figure 2.65.

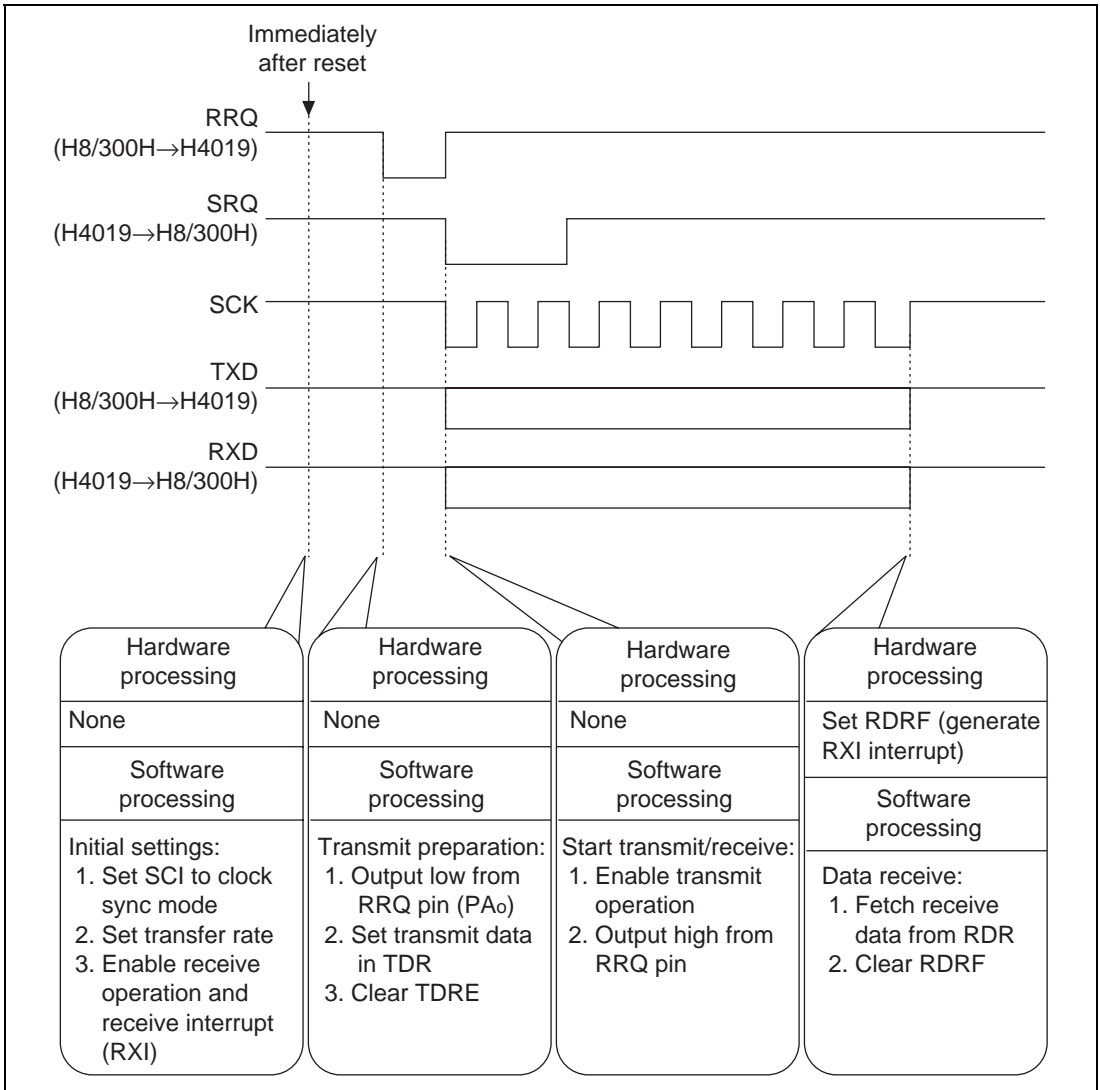


Figure 2.65 Principle of Simultaneous Transmit/Receive Operation

2.12.4 Description of Software

The software for simultaneous transmit/receive operation is described below.

Table 2.58 Description of Modules

Module Name	Label Name	Function
Main routine	CSCMN	SCI initial setting and transmit/receive management
Data receive end	CSCIRX	Conducts data transmission activated by RXI interrupt

Table 2.59 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
R6L	Sets data received from the console	1 byte	Data receive end	O
			Main routine	I
RXENDF	Flag indicating end of receive 1: receive end, 0: receiving	1 bit	Data receive end	O
			Main routine	I

Table 2.60 Description of Internal Registers

Register Name	Function	Module Name
SMR	Sets SCI mode (clock sync mode), transfer format and clock selection for baud rate generator (ϕ clock input)	Main routine
SCR	Enables interrupt (RXI) and sets SCI to transmit/receive	Main routine
SSR	Starts transmit due to clearing of TDRE (b7)	Main routine
RDR	Sets data received from H4019	Data receive end
TDR	Sets data for transfer to H4019	Main routine
BRR	Sets transfer rate	Main routine
PADDR	Sets PORT A I/O	Main routine
PADR	Operates PRQ pin and CTS pin	Main routine

Table 2.61 Description of General Registers

Module Name	Register Name	Function
Main routine	R0L	Used as work register during setting of SCI and argument RAM
	R6L	Used as argument for receive data acceptance
Data receive end	R6L	Used as receive data transfer argument

Table 2.62 Description of RAM

Label Name, Register Name	Function	Data Length	Module Name
RVDATA	Sets received data	1 byte	Main routine
TRDATA	Sets data for transmission	1 byte	Main routine

2.12.5 Flowcharts

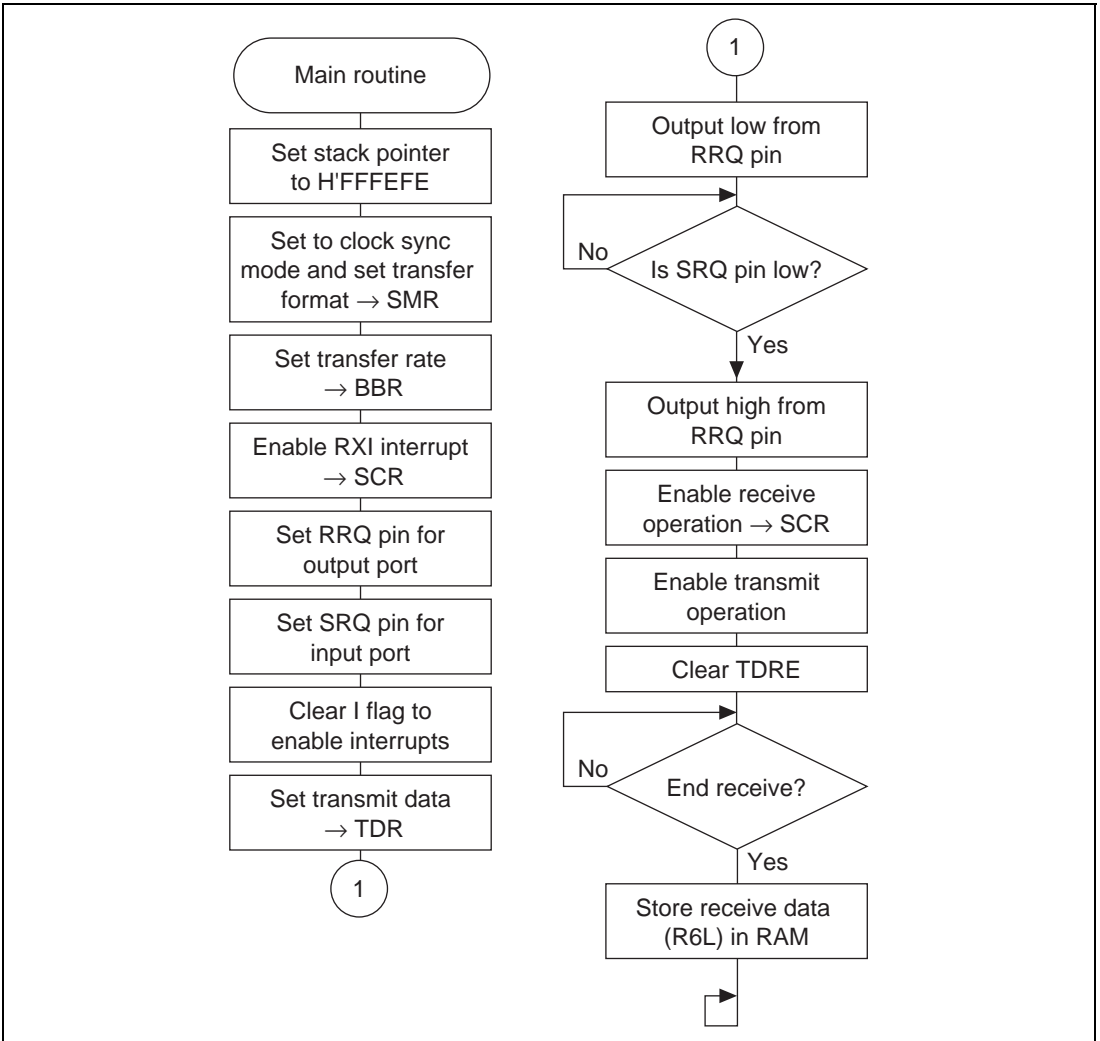


Figure 2.66 Main Routine Flowchart

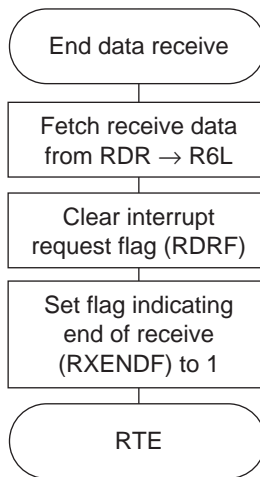


Figure 2.67 End Data Receive Flowchart

2.12.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION  VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES      .DATA.L   CSCMN
10         10 ;
11 00001C   11          .ORG      H'00001C
12 00001C 00010000  12 NMI     .DATA.L   CSCMN
13         13 ;
14 000030   14          .ORG      H'000030
15 000030 00010000  15 IRQ0   .DATA.L   CSCMN
16 000034 00010000  16 IRQ1   .DATA.L   CSCMN
17 000038 00010000  17 IRQ2   .DATA.L   CSCMN
18 00003C 00010000  18 IRQ3   .DATA.L   CSCMN
19 000040 00010000  19 IRQ4   .DATA.L   CSCMN
20 000044 00010000  20 IRQ5   .DATA.L   CSCMN
21 000048 00010000  21 IRQ6   .DATA.L   CSCMN
22 00004C 00010000  22 IRQ7   .DATA.L   CSCMN
23 000050 00010000  23 WOVI   .DATA.L   CSCMN
24 000054 00010000  24 CMI     .DATA.L   CSCMN
25         25 ;
26 000060   26          .ORG      H'000060
27 000060 00010000  27 IMIA0   .DATA.L   CSCMN
28 000064 00010000  28 IMIB0   .DATA.L   CSCMN
29 000068 00010000  29 OVI0    .DATA.L   CSCMN
30         30 ;
31 000070   31          .ORG      H'000070
32 000070 00010000  32 IMIA1   .DATA.L   CSCMN
33 000074 00010000  33 IMIB1   .DATA.L   CSCMN
34 000078 00010000  34 OVI1    .DATA.L   CSCMN
35         35 ;
36 000080   36          .ORG      H'000080
37 000080 00010000  37 IMIA2   .DATA.L   CSCMN
38 000084 00010000  38 IMIB2   .DATA.L   CSCMN
39 000088 00010000  39 OVI2    .DATA.L   CSCMN
40         40 ;
41 000090   41          .ORG      H'000090
42 000090 00010000  42 IMIA3   .DATA.L   CSCMN
43 000094 00010000  43 IMIB3   .DATA.L   CSCMN

```

```

44 000098 00010000      44 OVI3      .DATA.L    CSCMN
45                               45 ;
46 0000A0                               46          .ORG      H'0000A0
47 0000A0 00010000      47 IMIA4     .DATA.L    CSCMN
48 0000A4 00010000      48 IMIB4     .DATA.L    CSCMN
49 0000A8 00010000      49 OVI4     .DATA.L    CSCMN
50                               50 ;
51 0000B0                               51          .ORG      H'0000B0
52 0000B0 00010000      52 DEND0A   .DATA.L    CSCMN
53 0000B4 00010000      53 DEND0B   .DATA.L    CSCMN
54 0000B8 00010000      54 DEND1A   .DATA.L    CSCMN
55 0000BC 00010000      55 DEND1B   .DATA.L    CSCMN
56 0000C0 00010000      56 DEND2A   .DATA.L    CSCMN
57 0000C4 00010000      57 DEND2B   .DATA.L    CSCMN
58 0000C8 00010000      58 DEND3A   .DATA.L    CSCMN
59 0000CC 00010000      59 DEND3B   .DATA.L    CSCMN
60 0000D0 00010000      60 ERI0     .DATA.L    CSCMN
61 0000D4 00010044      61 RXI0     .DATA.L    CSCIRX
62 0000D8 00010000      62 TXI0     .DATA.L    CSCMN
63 0000DC 00010000      63 TEI0     .DATA.L    CSCMN
64 0000E0 00010000      64 ERI1     .DATA.L    CSCMN
65 0000E4 00010000      65 RXI1     .DATA.L    CSCMN
66 0000E8 00010000      66 TXI1     .DATA.L    CSCMN
67 0000EC 00010000      67 TEI1     .DATA.L    CSCMN
68 0000F0 00010000      68 ADI      .DATA.L    CSCMN
69                               69 ;
70                               70 ;*****
71                               71 ;* *
72                               72 ;*      RAM ALLOCATION *
73                               73 ;* *
74                               74 ;*****
75                               75 ;
76 FFFF00                               76          .SECTION  RAM,DATA,LOCATE=H'FFFF00
77                               77 ;
78 FFFF00 00000001      78 TRDATA   .RES.B    1          ;Transmit data
79 FFFF01 00000001      79 RVDATA   .RES.B    1          ;Receive data
80 FFFF02 00000001      80 FLAG     .RES.B    1          ;Flags
81          00000000      81 RXENDF   .EQU     0          ;Receive end flag
82                               82 ;
83                               83 ;*****
84                               84 ;* *
85                               85 ;*      SYMBOL DEFINITIONS *
86                               86 ;* *
87                               87 ;*****
88                               88 ;

```

```

89          89 ;*****          SCI ch0
*****
90          00FFFFB0      90 SMR0      .EQU      H'FFFFB0      ;Serial mode register
91          91 ;
92          00FFFFB1      92 BRR0      .EQU      H'FFFFB1 ;Bit rate register
93          93 ;
94          00FFFFB2      94 SCR0      .EQU      H'FFFFB2 ;srial control register
95          00000006      95 RIE      .EQU      6          ;Receive interrupt enable
96          00000005      96 TE      .EQU      5          ;Transmit enable
97          00000004      97 RE      .EQU      4          ;Receive enable
98          98 ;
99          00FFFFB3      99 TDR0      .EQU      H'FFFFB3 ;Transmit data register
100         100 ;
101         00FFFFB4      101 SSR0      .EQU      H'FFFFB4 ;Serial status register
102         00000007      102 TDRE      .EQU      7          ;Transmit data register empty
103         00000006      103 RDRF      .EQU      6          ;Receive data register full
104         104 ;
105         00FFFFB5      105 RDR0      .EQU      H'FFFFB5 ;Receive data register
106         106 ;
107         107 ;***** PORT A *****
108         108 ;
109         00FFFFD1      109 PADDR      .EQU      H'FFFFD1 ;Port A data direction
                                register
110         110 ;
111         00FFFFD3      111 PADR      .EQU      H'FFFFD3 ;Port A data register
112         00000000      112 RRQ      .EQU      0          ;RRQ bit
113         00000001      113 SRQ      .EQU      1          ;SRQ bit
114         114 ;
115         115 ;*****
116         116 ;*
117         117 ;*      MAIN PROGRAM : CSCMN
118         118 ;*
119         119 ;*****
120         120 ;
121 010000      121          .SECTION  PROG, CODE, LOCATE=H'010000
122         122 ;
123         00010000      123 CSCMN:    .EQU      $
124 010000      7A0700FFFEFE      124      MOV.L      #H'FFFEFE, SP      ;Set stack pointer
125 010006      F880          125      MOV.B      #B'10000000, R0L ;Clock mode, 8bit
126 010008      38B0          126      MOV.B      R0L, @SMR0
127 01000A      F803          127      MOV.B      #3, R0L          ;CKS0, 1=0, N=3
128 01000C      38B1          128      MOV.B      R0L, @BRR0      ;Then 1Mbps
129 01000E      7FB27060      129      BSET.B      #RIE, @SCR0      ;Enable RXI interrupt
130 010012      7FB27040      130      BSET.B      #RE, @SCR0      ;Enable receive
131 010016      F801          131      MOV.B      #H'01, R0L      ;SRQ is input port
132 010018      38D1          132      MOV.B      R0L, @PADDR      ;RRQ is output port

```



```

133                                     133 ;
134 01001A 067F                        134         ANDC     #B'01111111,CCR ;Clear I-FLAG
135 01001C 7FD37200                    135         BCLR.B   #RRQ,@PADR      ;RRQ is 'Low'
136 010020 2800                        136         MOV.B    @TRDATA,R0L      ;Set transmit data to
                                     ;TDR

137 010022 38B3                        137         MOV.B    R0L,@TDR0
138 010024 7FB47270                    138         BCLR.B   #TDRE,@SSR0      ;Start transmit
139 010028                              139 CSCMN01:
140 010028 7ED37310                    140         BTST.B   #SRQ,@PADR      ;SRQ is 'Low' ?
141 01002C 46FA                        141         BNE     CSCMN01          ;No
142                                     142 ;
143 01002E 7FD37000                    143         BSET.B   #PRQ,@PADR      ;RRQ is 'High'
144 010032 7FB27050                    144         BSET.B   #TE,@SCR0       ;Enable transmit
145                                     145 ;
146 010036                              146 CSCMN02:
147 010036 7E027300                    147         BTST.B   #RXENDF,@FLAG    ;Receive complete ?
148 01003A 47FA                        148         BEQ     CSCMN02          ;No
149                                     149 ;
150 01003C 7F027200                    150         BCLR.B   #RXENDF,@FLAG    ;Clear RXENDF
151 010040 3E01                        151         MOV.B    R6L,@RVDATA      ;Set receive data
152 010042                              152 CSCMN99:
153 010042 40FE                        153         BRA     CSCMN99
154                                     154 ;
155                                     155 ;*****
156                                     156 ;*                                     *
157                                     157 ;*          NAME : CSCIRX(STOP COUNTER)      *
158                                     158 ;*                                     *
159                                     159 ;*****
160                                     160 ;                                     *
161                                     161 ;*          ENTRY : NOTHING                    *
162                                     162 ;*          RETURNS : TSTR(TIMER START REG)   *
163                                     163 ;*                                     *
164                                     164 ;*****
165                                     165 ;
166                                     166 CSCIRX:  .EQU     $
167 010044 2EB5                        167         MOV.B    @RDR0,R6L      ;Get receive data to
R6L
168 010046 7FB47260                    168         BCLR.B   #RDRF,@SSR0      ;Clear RDRF
169 01004A 7F027000                    169         BSET.B   #RXENDF,@FLAG    ;Set RXENDF
170 01004E 5670                        170         RTE
171                                     171 ;
172                                     172         .END

*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```

2.13 Multiprocessor Communication

MCU: H8/3003

Function: SCI (multiprocessor communication)

2.13.1 Specifications

- As shown in figure 2.68, several H8/3003 devices are linked on a common communication line conducting data transmit and receive
- The master H8/3003 transmits to two slave side H8/3003 devices. The slave side H8/3003 receive only data sent to their own stations
- Transmit/receive format: 9600 bps, 8-bit data, 1 stop bit and non-parity bit

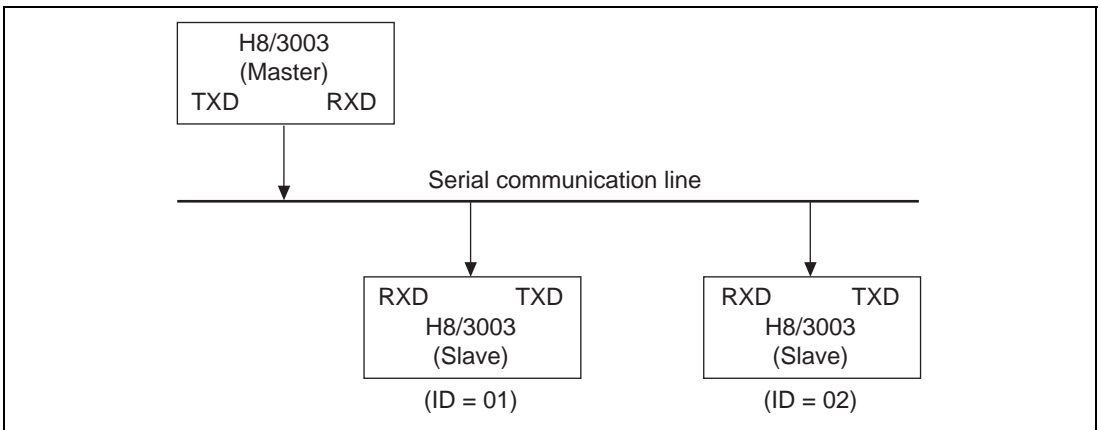


Figure 2.68 Block Diagram of Asynchronous SCI Interface Using Multiprocessor Functions

2.13.2 Description of Functions Used

In this sample task, the SCI's multiprocessor communication functions are used for communication between several H8/3003 devices.

Figure 2.69 shows a block diagram of the master side SCI used in the sample task. The following SCI functions are employed:

- Asynchronous mode data communication synchronized in character units (asynchronous mode)
- Communication with format having an added multiprocessor bit (multiprocessor communication function)
- Interrupt generation at start of transmit (TXI interrupt)
- Interrupt generation at end of transmit (TEI interrupt)

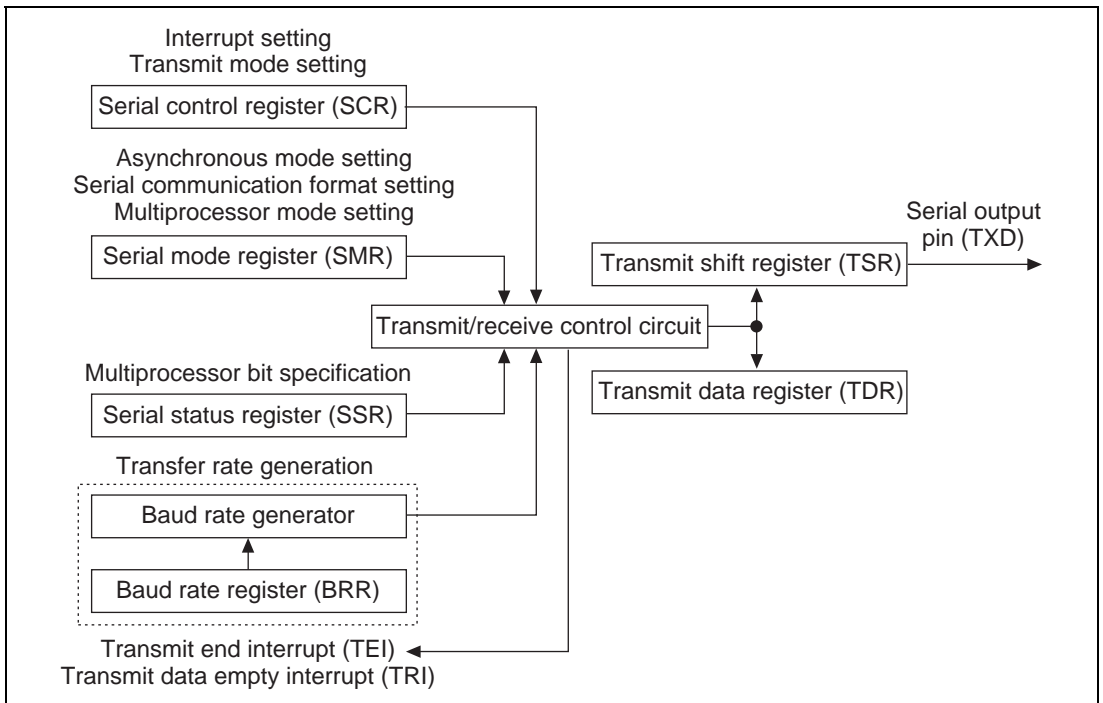


Figure 2.69 Block Diagram of Master Side SCI

Figure 2.70 shows a block diagram of the slave side SCI used in the sample task. The following SCI functions are used:

- Asynchronous mode data communication synchronized in character units (asynchronous mode)
- Communication with format having an added multiprocessor bit (multiprocessor communication function)
- Interrupt generation on reception of multiprocessor bit (multiprocessor interrupt)
- Interrupt generation at end of receive (RXI interrupt)

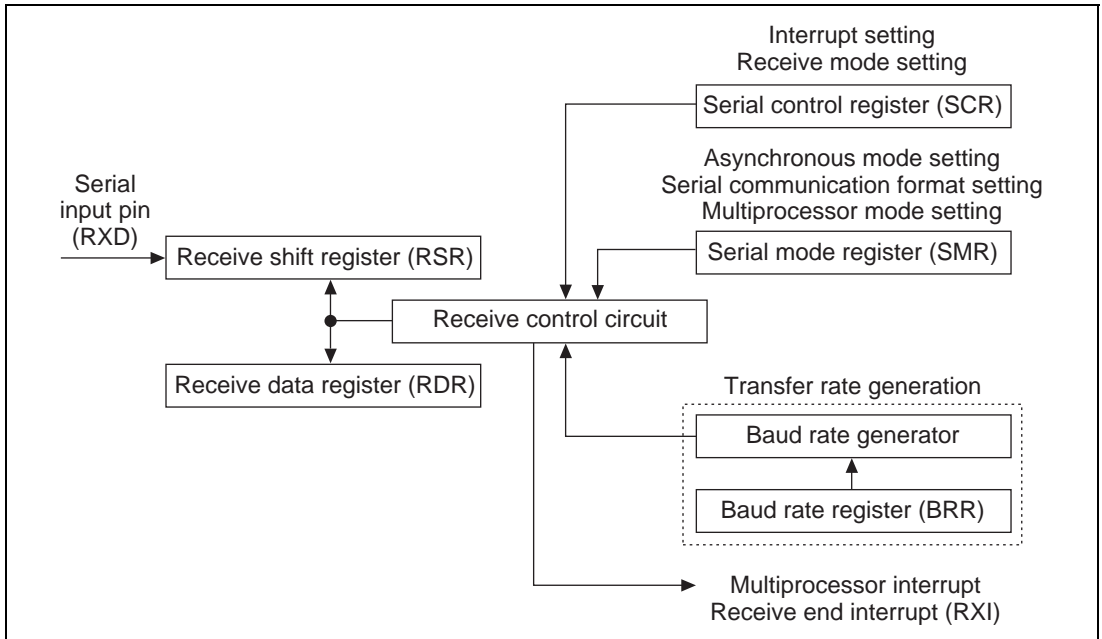


Figure 2.70 Block Diagram of Slave Side SCI

Table 2.63 shows the SCI function allocation for multiprocessor communication in this sample task.

Table 2.63 SCI Function Allocation

SCI Function	Function
RXD	Receive data from H8/3003
TXD	Transmit data to H8/3003
SMR	Sets SCI in asynchronous mode and multiprocessor mode
SCR	Enables transmit/receive interrupts, and sets SCI to transmit/receive mode
SSR	Sets to starts transmit/multiprocessor bit
RDR	Setting for receive data from H8/3003
TDR	Setting for transmit data to H8/3003
BRR	Sets transfer rate

2.13.3 Description of Operations

Master Side Operation: Figure 2.71 shows the principle of operation on the master side. Data is transmitted to the receive station H8/3003 due to hardware and software processing with the timing shown in figure 2.71.

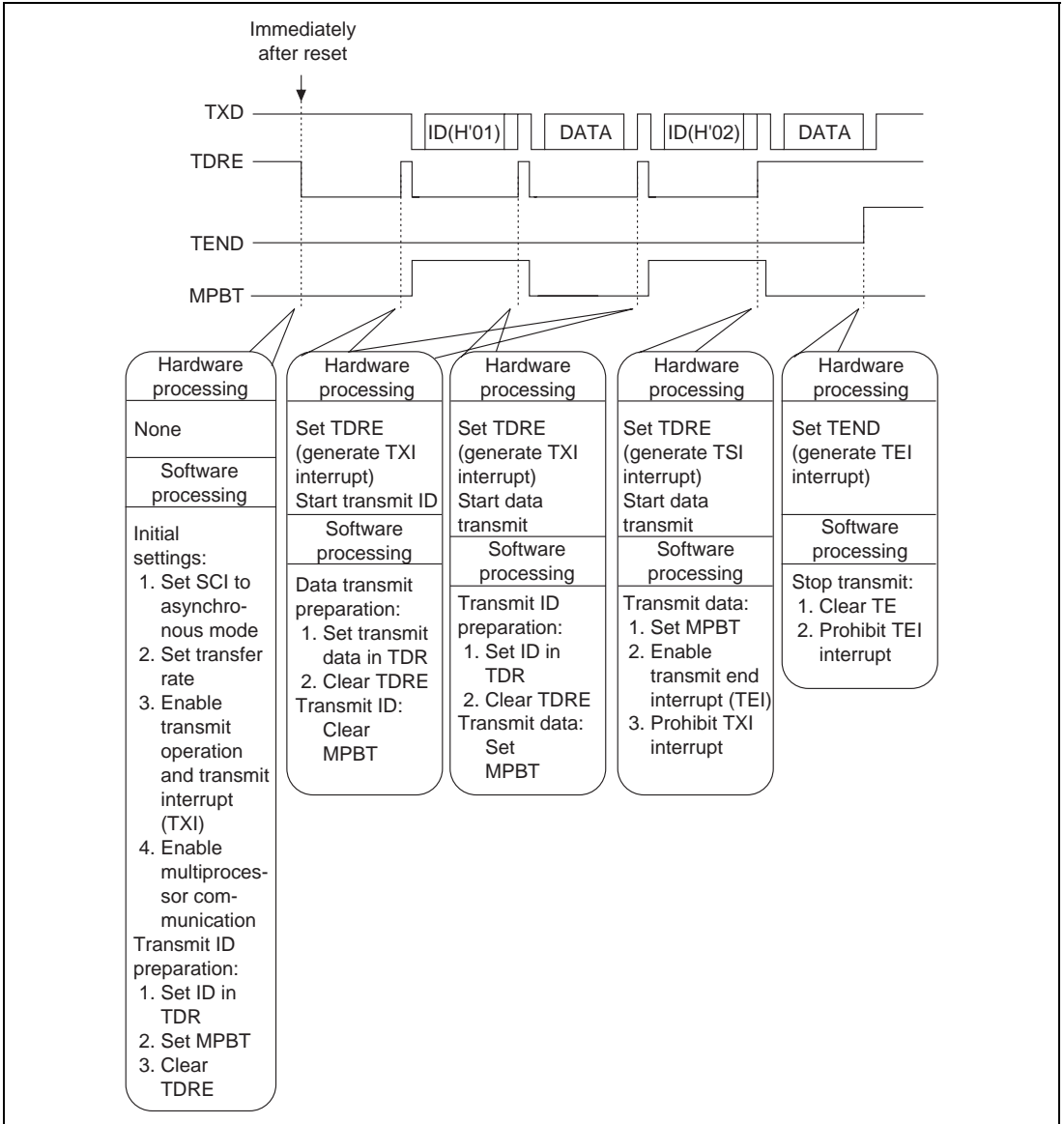


Figure 2.71 Principle of Multiprocessor Communication Operation (Master Side)

Slave Side Operation: Figure 2.72 shows the principle of operation on the slave side. Data is received from the transmit station due to hardware and software processing with the timing shown in figure 2.72.

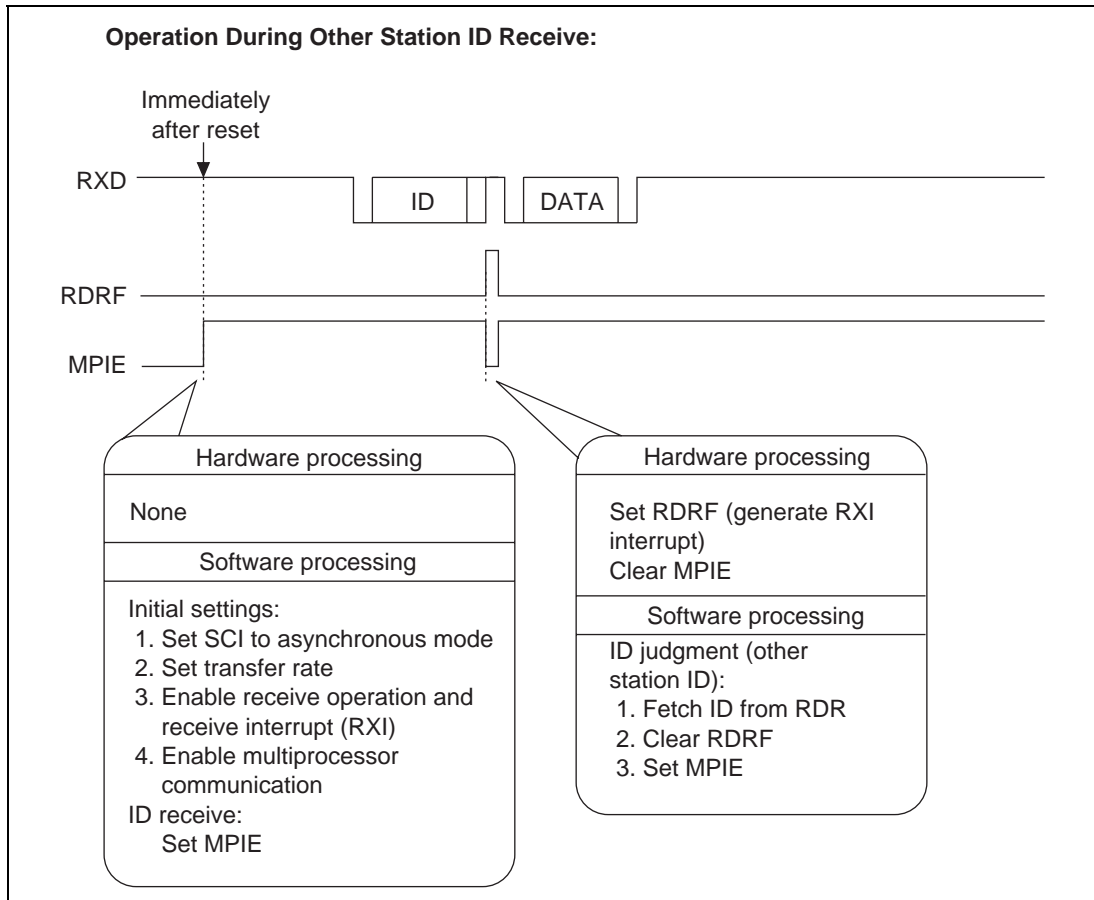


Figure 2.72 Principle of Multiprocessor Communication Operation (Slave Side) (1)

Operation During Own Station ID Receive:

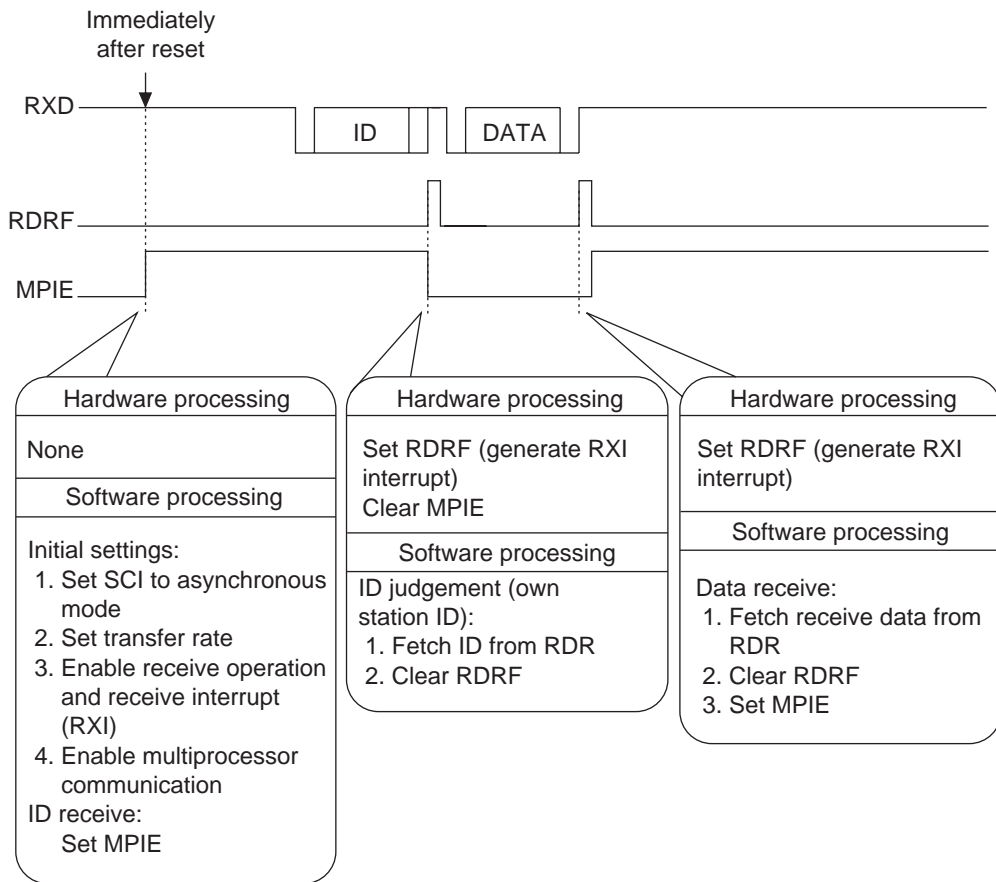


Figure 2.72 Principle of Multiprocessor Communication Operation (Slave Side) (2)

2.13.4 Description of Software

Description of Master Software: The master software for multiprocessor communication is described below.

Table 2.64 Description of Modules

Module Name	Label Name	Function
Main routine	MPMASMN	Initial setting of SCI
Data transmit	MPSCITX	ID and data transmit activated by TXI interrupt
Transmit end	MPSCITE	Stops SCI transmit activated by TEI interrupt

Table 2.65 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
TXDATA	Buffer for storage of ID and data for transmission to slave side H8/3003	4 bytes	Main routine	O
			Data transmit	I

Table 2.66 Description of Internal Registers

Register Name	Function	Module Name
SMR	Sets SCI mode (asynchronous mode), transfer format and clock selection for baud rate generator (ϕ clock input)	Main routine
SCR	Enables interrupts (TXI, TEI) and sets SCI transmit enable/disable	Main routine
		Transfer end
SSR	Starts transmit due to clearing of TDRE (b7)	Main routine
		Data transmit
TDR	Sets ID and data for transmission to slave side H8/3003	Main routine
		Data transmit
BRR	Sets transfer rate	Main routine

Table 2.67 Description of General Registers

Module Name	Register Name	Function
Main routine	ROL	Used as work register during setting of SCI and argument RAM
Data transmit	ROL	Used as work register during setting of data to TDR

Table 2.68 Description of RAM

Label Name	Module Name	Data Length	Function
TXCNT	Data transmit	1 byte	Counts transmitted data volume

Description of Slave Software: The slave software for multiprocessor communication is described below.

Table 2.69 Description of Modules

Module Name	Label Name	Function
Main routine	MPSRVMN	Initial setting of SCI
Data receive	MPSCIRX	ID and data receive activated by RXI interrupt

Table 2.70 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
R6L	Sets received ID and data	1 byte	Data receive	O
			Main routine	I
IDRCVF	Flag indicating own station ID receive 1: ID received, 0: no ID received	1 bit	Data receive	O
			Main routine	I
DTRCVF	Flag indicating data receive 1: data received, 0: no data received	1 bit	Data receive	O
			Main routine	I

Table 2.71 Description of Internal Registers

Register Name	Function	Module Name
SMR	Sets SCI mode (asynchronous mode), transfer format and clock selection for baud rate generator (ϕ clock input)	Main routine
SCR	Enables interrupts (RXI) and sets SCI to receive enable	Main routine
RDR	Sets ID and data received from master side H8/3003	Data receive
BRR	Sets transfer rate	Main routine

Table 2.72 Description of General Registers

Module Name	Register Name	Function
Main routine	R0L	Used as work register during setting of SCI and argument RAM
	R6L	Used as acceptance argument for received data
Data receive	R6L	Used as argument for transfer of receive data to main routine

Table 2.73 Description of RAM

Label Name	Module Name	Data Length	Function
RXID	Main routine	1 byte	Sets received ID
RXDATA	Main routine	1 byte	Sets received data
MYID	Data receive	1 byte	Sets own station ID

2.13.5 Flowcharts

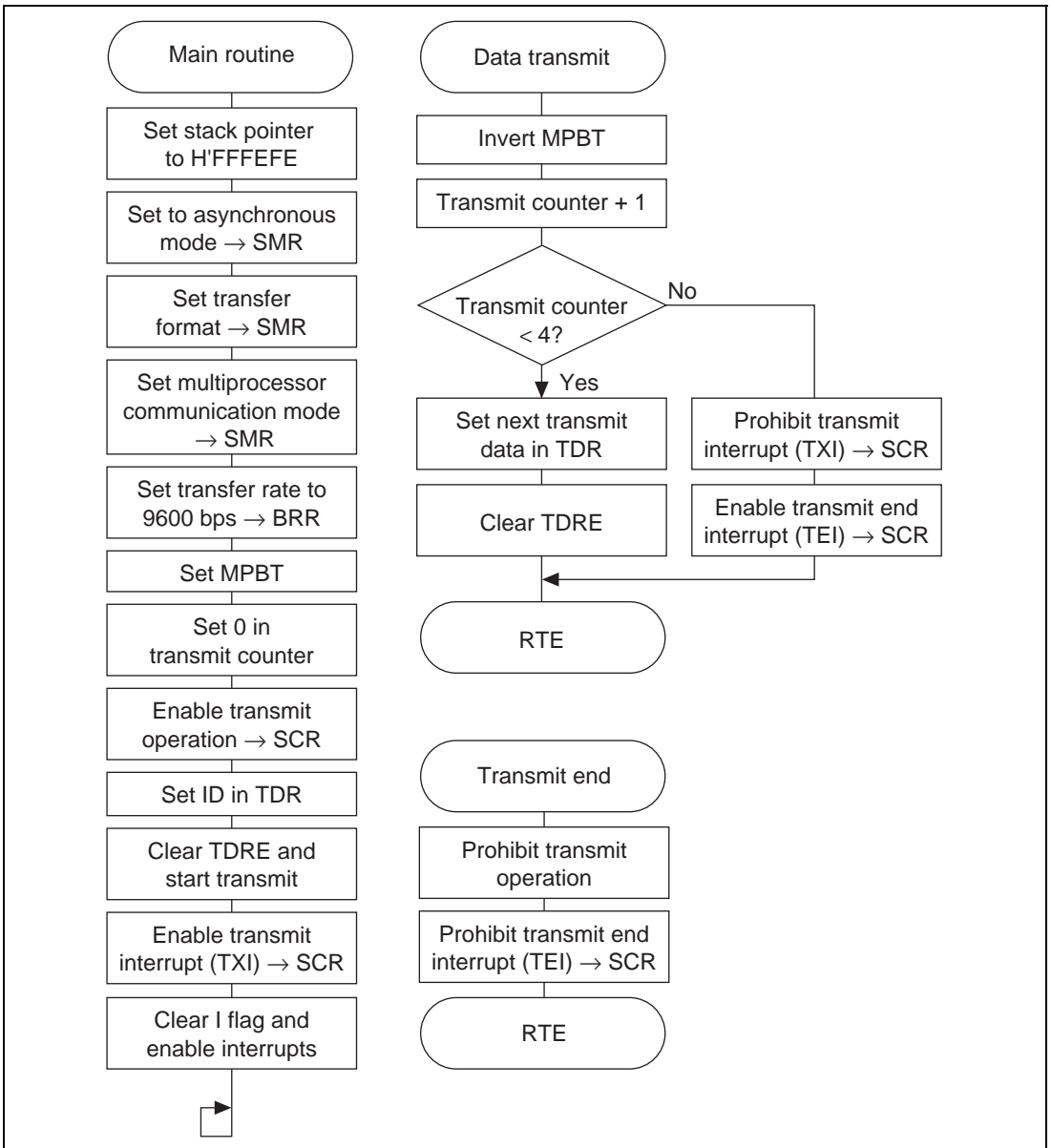


Figure 2.73 Flowchart of Master Side Software

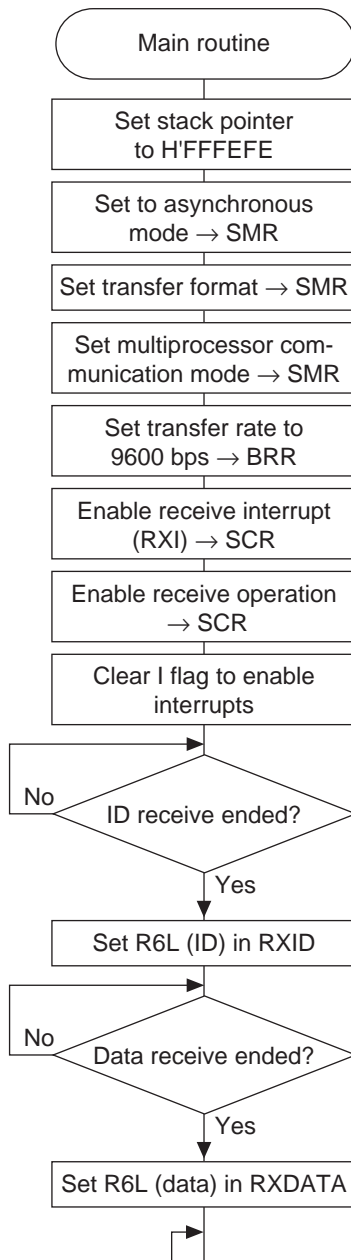


Figure 2.74 Flowchart of Slave Side Software (Main Routine)

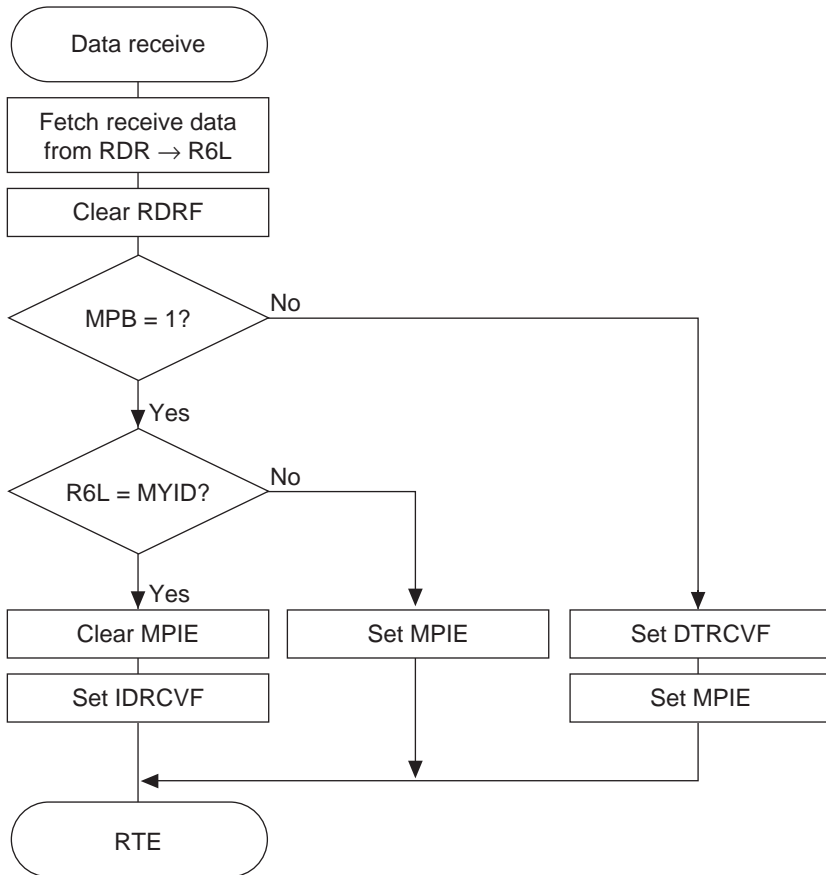


Figure 2.75 Flowchart of Slave Side Software (Data Receive)

2.13.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7          7          .CPU          300HA
8 000000   8          .SECTION    VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L    MPMASMN
11         11 ;
12 00001C   12          .ORG          H'00001C
13 00001C 00010000 13 NMI          .DATA.L    MPMASMN
14 000020 00010000 14 TRAP0        .DATA.L    MPMASMN
15 000024 00010000 15 TRAP1        .DATA.L    MPMASMN
16 000028 00010000 16 TRAP2        .DATA.L    MPMASMN
17 00002C 00010000 17 TRAP3        .DATA.L    MPMASMN
18         18 ;
19 000030   19          .ORG          H'000030
20 000030 00010000 20 IRQ0         .DATA.L    MPMASMN
21 000034 00010000 21 IRQ1         .DATA.L    MPMASMN
22 000038 00010000 22 IRQ2         .DATA.L    MPMASMN
23 00003C 00010000 23 IRQ3         .DATA.L    MPMASMN
24 000040 00010000 24 IRQ4         .DATA.L    MPMASMN
25 000044 00010000 25 IRQ5         .DATA.L    MPMASMN
26 000048 00010000 26 IRQ6         .DATA.L    MPMASMN
27 00004C 00010000 27 IRQ7         .DATA.L    MPMASMN
28 000050 00010000 28 WOVI        .DATA.L    MPMASMN
29 000054 00010000 29 CMI          .DATA.L    MPMASMN
30         30 ;
31 000060   31          .ORG          H'000060
32 000060 00010000 32 IMIA0        .DATA.L    MPMASMN
33 000064 00010000 33 IMIB0        .DATA.L    MPMASMN
34 000068 00010000 34 OVI0         .DATA.L    MPMASMN
35         35 ;
36 000070   36          .ORG          H'000070
37 000070 00010000 37 IMIA1        .DATA.L    MPMASMN
38 000074 00010000 38 IMIB1        .DATA.L    MPMASMN
39 000078 00010000 39 OVI1         .DATA.L    MPMASMN
40         40 ;
41 000080   41          .ORG          H'000080
42 000080 00010000 42 IMIA2        .DATA.L    MPMASMN
43 000084 00010000 43 IMIB2        .DATA.L    MPMASMN

```

```

44 000088 00010000      44 OVI2      .DATA.L      MPMASMN
45                               45 ;
46 000090                               46          .ORG          H'000090
47 000090 00010000      47 IMIA3      .DATA.L      MPMASMN
48 000094 00010000      48 IMIB3      .DATA.L      MPMASMN
49 000098 00010000      49 OVI3      .DATA.L      MPMASMN
50                               50 ;
51 0000A0                               51          .ORG          H'0000A0
52 0000A0 00010000      52 IMIA4      .DATA.L      MPMASMN
53 0000A4 00010000      53 IMIB4      .DATA.L      MPMASMN
54 0000A8 00010000      54 OVI4      .DATA.L      MPMASMN
55                               55 ;
56 0000B0                               56          .ORG          H'0000B0
57 0000B0 00010000      57 DEND0A     .DATA.L      MPMASMN
58 0000B4 00010000      58 DEND0B     .DATA.L      MPMASMN
59 0000B8 00010000      59 DEND1A     .DATA.L      MPMASMN
60 0000BC 00010000      60 DEND1B     .DATA.L      MPMASMN
61 0000C0 00010000      61 DEND2A     .DATA.L      MPMASMN
62 0000C4 00010000      62 DEND2B     .DATA.L      MPMASMN
63 0000C8 00010000      63 DEND3A     .DATA.L      MPMASMN
64 0000CC 00010000      64 DEND3B     .DATA.L      MPMASMN
65 0000D0 00010000      65 ERI0       .DATA.L      MPMASMN
66 0000D4 00010000      66 RXI0       .DATA.L      MPMASMN
67 0000D8 0001003A      67 TXI0       .DATA.L      MPSCITY
68 0000DC 00010070      68 TEI0       .DATA.L      MPSCITE
69 0000E0 00010000      69 ERI1       .DATA.L      MPMASMN
70 0000E4 00010000      70 RXI1       .DATA.L      MPMASMN
71 0000E8 00010000      71 TXI1       .DATA.L      MPMASMN
72 0000EC 00010000      72 TEI1       .DATA.L      MPMASMN
73 0000F0 00010000      73 ADI        .DATA.L      MPMASMN
74                               74 ;
75                               75 ;*****
76                               76 ;*                                     *
77                               77 ;*          RAM ALLOCATION          *
78                               78 ;*                                     *
79                               79 ;*****
80                               80 ;
81 FFFF00                               81          .SECTION   RAM,DATA,LOCATE=H'FFFF00
82                               82 ;
83 FFFF00 00000004      83 TXDATA     .RES.B      4          ;Transmit data
84 FFFF04 00000001      84 TXCNT      .RES.B      1          ;Transmit counter
85                               85 ;
86                               86 ;*****
87                               87 ;*                                     *
88                               88 ;*          SYMBOL DEFINITIONS          *
89                               89 ;*                                     *

```



```

90 ;*****
91 ;
92 00FFFFB0 92 SMR0 .EQU H'FFFFB0 ;Serial mode register
93 ;
94 00FFFFB1 94 BRR0 .EQU H'FFFFB1 ;Bit rate register
95 ;
96 00FFFFB2 96 SCR0 .EQU H'FFFFB2 ;Serial control register
97 00000007 97 TIE .EQU 7 ;Transmit interrupt enable
98 00000005 98 TE .EQU 5 ;Transmit enable
99 00000002 99 TEIE .EQU 2 ;Transmit end interrupt
enable
100 ;
101 00FFFFB3 101 TDR0 .EQU H'FFFFB3 ;Transmit data register
102 ;
103 00FFFFB4 103 SSR0 .EQU H'FFFFB4 ;Serial status register
104 00000007 104 TDRE .EQU 7 ;Transmit data register
;empty
105 00000000 105 MPBT .EQU 0 ;Multi processor bit
;transfer
106 ;
107 ;*****
108 ;* *
109 ;* MAIN PROGRAM : MPMASMN *
110 ;* *
111 ;*****
112 ;
113 010000 113 .SECTION PROG, CODE, LOCATE=H'010000
114 ;
115 00010000 115 MPMASMN: .EQU $
116 010000 7A0700FFFEFE 116 MOV.L #H'FFFEFE, SP ;Set stack pointer
117 010006 F804 117 MOV.B #B'00000100, R0L ;Asyncre mode, 8bit, np,
;stop1, multi processer
;m
118 010008 38B0 118 MOV.B R0L, @SMR0
119 01000A F833 119 MOV.B #51, R0L ;CKS0, 1=0, N=51
120 01000C 38B1 120 MOV.B R0L, @BRR0 ;Then 9600bps
121 01000E 7FB47000 121 BSET.B #MPBT, @SSR0 ;Set MPBT
122 ;
123 010012 790001AA 123 MOV.W #H'01AA, R0 ;Set ID=01 & data
124 010016 6B80FF00 124 MOV.W R0, @TXDATA+0
125 01001A 79000255 125 MOV.W #H'0255, R0 ;Set ID=02 & data
126 01001E 6B80FF02 126 MOV.W R0, @TXDATA+2
127 ;
128 010022 F800 128 MOV.B #H'0, R0L ;Set TXCNT=0
129 010024 3804 129 MOV.B R0L, @TXCNT
130 ;

```

```

131 010026 2800          131      MOV.B      @TXDATA,R0L      ;Set ID(H'01) to TDR0
132 010028 3883          132      MOV.B      R0L,@TDR0
133 01002A 7FB27050      133      BSET.B     #TE,@SCR0      ;Enable transmit
134 01002E 7FB47270      134      BCLR.B     #TDRE,@SSR0    ;Start transmit
135 010032 7FB27070      135      BSET.B     #TIE,@SCR0    ;Enable TXI interrupt
136                                136      ;
137 010036 067F          137      ANDC      #B'01111111,CCR ;Clear I-FLAG
138 010038                                138 MPMAS99:
139 010038 40FE          139      BRA      MPMAS99
140                                140      ;
141                                141      ;*****
142                                142      ;*                                *
143                                143      ;*      NAME : MPSCITX(SCI transmit interrupt) *
144                                144      ;*                                *
145                                145      ;*****
146                                146      ;*                                *
147                                147      ;*      ENTRY : TXDATA (Transmit data) *
148                                148      ;*      RETURNS : NOTHING *
149                                149      ;*                                *
150                                150      ;*****
151                                151      ;
152                                152 MPSCITX: .EQU      $
153 01003A 6DF0          153      PUSH.W   R0              ;Save registers
154 01003C 6DF1          154      PUSH.W   R1
155                                155      ;
156 01003E 7FB47100      156      BNOT.B   #MPBT,@SSR0   ;Invert MPBT
157 010042 F100          157      MOV.B      #0,R1H          ;TXCNT = TXCNT + 1
158 010044 2904          158      MOV.B      @TXCNT,R1L
159 010046 0A09          159      INC      R1L
160 010048 3904          160      MOV.B      R1L,@TXCNT
161 01004A A904          161      CMP.B      #4,R1L          ;TXCNT >= 4 ?
162 01004C 58400012      162      BCC      MPTX10          ;Yes
163                                163      ;
164 010050 78106A2800FFFF00 164 MOV.B      @(TXDATA,R1),R0L ;load next trans data
165 010058 38B3          165      MOV.B      R0L,@TDR0      ;Set next data to TDR0
166 01005A 7FB47270      166      BCLR.B     #TDRE,@SSR0    ;Start transmit
167 01005E 58000008      167      BRA      MPTX99
168                                168      ;
169 010062                                169 MPTX10:
170 010062 7FB27270      170      BCLR.B     #TIE,@SCR0    ;Disable TXI
171 010066 7FB27020      171      BSET.B     #TEIE,@SCR0    ;Enable TEI interrupt
172                                172      ;
173 01006A                                173 MPTX99:
174 01006A 6D71          174      POP.W    R1
175 01006C 6D70          175      POP.W    R0
176 01006E 5670          176      RTE

```

```

177 ;
178 ;*****
179 ;*
180 ;* NAME : MPSCITI(SCI transmit end interrupt) *
181 ;*
182 ;*****
183 ;*
184 ;* ENTRY : NOTHING *
185 ;* RETURNS : NOTHING *
186 ;*
187 ;*****
188 ;
189 MPSCITE: .EQU $
190 BCLR.B #TE,@SCR0 ;Disable transmit
191 BCLR.B #TEIE,@SCR0 ;disable TEI interrupt
192 RTE
193 ;
194 .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0

```

```

1      1 ;*****
2      2 ;*
3      3 ;*      VECTOR ADDRESS
4      4 ;*
5      5 ;*****
6      6 ;
7      7      .CPU      300HA
8      8      .SECTION  VECT, CODE, LOCATE=H'000000
9      9 ;
10     10 RES      .DATA.L  MPSRVMN
11     11 ;
12     12      .ORG      H'00001C
13     13 NMI      .DATA.L  MPSRVMN
14     14 ;
15     15      .ORG      H'000030
16     16 IRQ0     .DATA.L  MPSRVMN
17     17 IRQ1     .DATA.L  MPSRVMN
18     18 IRQ2     .DATA.L  MPSRVMN
19     19 IRQ3     .DATA.L  MPSRVMN
20     20 IRQ4     .DATA.L  MPSRVMN
21     21 IRQ5     .DATA.L  MPSRVMN
22     22 IRQ6     .DATA.L  MPSRVMN
23     23 IRQ7     .DATA.L  MPSRVMN
24     24 WOVI     .DATA.L  MPSRVMN
25     25 CMI      .DATA.L  MPSRVMN
26     26 ;
27     27      .ORG      H'000060
28     28 IMIA0    .DATA.L  MPSRVMN
29     29 IMIB0    .DATA.L  MPSRVMN
30     30 OVI0     .DATA.L  MPSRVMN
31     31 ;
32     32      .ORG      H'000070
33     33 IMIA1    .DATA.L  MPSRVMN
34     34 IMIB1    .DATA.L  MPSRVMN
35     35 OVI1     .DATA.L  MPSRVMN
36     36 ;
37     37      .ORG      H'000080
38     38 IMIA2    .DATA.L  MPSRVMN
39     39 IMIB2    .DATA.L  MPSRVMN
40     40 OVI2     .DATA.L  MPSRVMN
41     41 ;
42     42      .ORG      H'000090
43     43 IMIA3    .DATA.L  MPSRVMN
44     44 IMIB3    .DATA.L  MPSRVMN
45     45 OVI3     .DATA.L  MPSRVMN
46     46 ;

```

```

47 0000A0          47          .ORG          H'0000A0
48 0000A0 00010000 48 IMIA4      .DATA.L      MPSRVMMN
49 0000A4 00010000 49 IMIB4      .DATA.L      MPSRVMMN
50 0000A8 00010000 50 OVI4       .DATA.L      MPSRVMMN
51                51 ;
52 0000B0          52          .ORG          H'0000B0
53 0000B0 00010000 53 DEND0A     .DATA.L      MPSRVMMN
54 0000B4 00010000 54 DEND0B     .DATA.L      MPSRVMMN
55 0000B8 00010000 55 DEND1A     .DATA.L      MPSRVMMN
56 0000BC 00010000 56 DEND1B     .DATA.L      MPSRVMMN
57 0000C0 00010000 57 DEND2A     .DATA.L      MPSRVMMN
58 0000C4 00010000 58 DEND2B     .DATA.L      MPSRVMMN
59 0000C8 00010000 59 DEND3A     .DATA.L      MPSRVMMN
60 0000CC 00010000 60 DEND3B     .DATA.L      MPSRVMMN
61 0000D0 00010000 61 ERI0       .DATA.L      MPSRVMMN
62 0000D4 00010032 62 RXI0       .DATA.L      MPSCIRX
63 0000D8 00010000 63 TXI0       .DATA.L      MPSRVMMN
64 0000DC 00010000 64 TEI0       .DATA.L      MPSRVMMN
65 0000E0 00010000 65 ERI1       .DATA.L      MPSRVMMN
66 0000E4 00010000 66 RXI1       .DATA.L      MPSRVMMN
67 0000E8 00010000 67 TXI1       .DATA.L      MPSRVMMN
68 0000EC 00010000 68 TEI1       .DATA.L      MPSRVMMN
69 0000F0 00010000 69 ADI        .DATA.L      MPSRVMMN
70                70 ;
71                71 ;*****
72                72 ;* *
73                73 ;*      RAM ALLOCATION *
74                74 ;* *
75                75 ;*****
76                76 ;
77 FFFF00          77          .SECTION   RAM,DATA,LOCATE=H'FFFF00
78                78 ;
79 FFFF00 00000001 79 RXID       .RES.B      1          ;Receive ID cord
80          00000001 80 MYID       .EQU        H'01      ;My ID cord
81                81 ;
82 FFFF01 00000001 82 RXDATA     .RES.B      1          ;Receive data
83                83 ;
84 FFFF02 00000001 84 RCVFLAG    .RES.B      1          ;Flag area
85          00000000 85 IDRCVF     .EQU        0          ;ID cord compare flag
86          00000001 86 DTRCVF     .EQU        1          ;Data receive flag
87                87 ;
88                88 ;*****
89                89 ;* *
90                90 ;*      SYMBOL DEFINITIONS *
91                91 ;* *
92                92 ;*****

```

```

93          93 ;
94          00FFFFFFB0      94 SMR0      .EQU      H'FFFFFFB0      ;Serial mode register
95
96          00FFFFFFB1      96 BRR0      .EQU      H'FFFFFFB1      ;Bit rate register
97
98          00FFFFFFB2      98 SCR0      .EQU      H'FFFFFFB2      ;Sirial control register
99          00000006      99 RIE      .EQU      6      ;Receive interrupt enable
100         00000004      100 RE      .EQU      4      ;Receive enable
101         00000003      101 MPIE      .EQU      3      ;Multi processor interrupt
;enable
102
103         00FFFFFFB3      103 TDR0      .EQU      H'FFFFFFB3      ;Transmit data register
104
105         00FFFFFFB4      105 SSR0      .EQU      H'FFFFFFB4      ;Serial status register
106         00000006      106 RDRF      .EQU      6      ;Receive data register full
107         00000002      107 TEND      .EQU      2      ;Transmit end
108         00000001      108 MPB      .EQU      1      ;Multi processor bit
109
110         00FFFFFFB5      110 RDR0      .EQU      H'FFFFFFB5      ;Receive data register
111
112         112 ;*****
113         113 ;*
114         114 ;*      MAIN PROGRAM : MPSRVMN
115         115 ;*
116         116 ;*****
117
118         010000      118          .SECTION      PROG,CODE,LOCATE=H'010000
119
120         00010000      120 MPSRVMN: .EQU      $
121         010000      7A0700FFFEFE      121          MOV.L      #H'FFFEFE,SP      ;Set stack pointer
122         010006      F804      122          MOV.B      #B'00000100,R0L      ;Asynco mode,8bit,np,
;stop1,multi processor
;m
123         010008      38B0      123          MOV.B      R0L,@SMR0
124         01000A      F833      124          MOV.B      #51,R0L      ;CKS0,1=0,N=51
125         01000C      38B1      125          MOV.B      R0L,@BRR0      ;Then 9600bps
126         01000E      7FB27060      126          BSET      #RIE,@SCR0      ;Enable RXI interrupt
127         010012      7FB27040      127          BSET      #RE,@SCR0      ;Enable receive
128
129         010016      067F      129          ANDC      #B'01111111,CCR ;Clear C-FLAG
130         010018      130 MPSRV01:
131         010018      7E027300      131          BTST      #IDRCVF,@RCVFLAG;ID receive ?
132         01001C      47FA      132          BEQ      MPSRV01      ;No
133
134         01001E      7F027200      134          BCLR      #IDRCVF,@RCVFLAG;Clear flag
135         010022      3E00      135          MOV.B      R6L,@RXID      ;Store receive ID

```

```

136                                     136 ;
137 010024                             137 MPSRV02:
138 010024 7E027310                     138         BTST      #DTRCVF,@RCVFLAG;DATA receive ?
139 010028 47FA                           139         BEQ       MPSRV02      ;No
140                                     140 ;
141 01002A 7F027210                       141         BCLR      #DTRCVF,@RCVFLAG;Clear flag
142 01002E 3E01                            142         MOV.B     R6L,@RXDATA    ;Store receive DATA
143 010030                             143 MPSRV99:
144 010030 40FE                           144         BRA       MPSRV99
145                                     145 ;
146                                     146 ;*****
147                                     147 ;       SCI RX INTERRUPT
148                                     148 ;*****
149         00010032                       149 MPSCIRX: .EQU      $
150 010032 6DF0                           150         PUSH.W    R0              ;Save registers
151 010034 6DF1                           151         PUSH.W    R1
152                                     152 ;
153 010036 2EB5                            153         MOV.B     @RDR0,R6L      ;Load receive data
154 010038 7FB47260                       154         BCLR      #RDRF,@SSR0   ;Clear RDRF
155                                     155 ;
156 01003C 7EB47310                       156         BTST      #MPB,@SSR0   ;MPB = 1 ?
157 010040 5870001A                       157         BEQ       MPRX02      ;No
158                                     158 ;
159 010044 AE01                            159         CMP.B     #MYID,R6L      ;Receive ID = My ID ?
160 010046 5860000C                       160         BNE      MPRX01      ;No
161                                     161 ;
162 01004A 7FB27230                       162         BCLR      #MPIE,@SCR0   ;Clear MPIE
163 01004E 7F027000                       163         BSET      #IDRCVF,@RCVFLAG;Set IDRCVF
164 010052 58000010                       164         BRA       MPRX99      ;Exit
165                                     165 ;
166 010056                             166 MPRX01:
167 010056 7FB27030                       167         BSET      #MPIE,@SCR0   ;Set MPIE
168 01005A 58000008                       168         BRA       MPRX99      ;Exit
169                                     169 ;
170 01005E                             170 MPRX02:
171 01005E 7F027010                       171         BSET      #DTRCVF,@RCVFLAG;Set DTRCVF
172 010062 7FB27030                       172         BSET      #MPIE,@SCR0   ;Set MPIE
173                                     173 ;
174 010066                             174 MPRX99:
175 010066 6D71                            175         POP.W     R1
176 010068 6D70                            176         POP.W     R0
177 01006A 5670                            177         RTE
178                                     178 ;
179                                     179         .END

*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```

2.14 Block Transfer

MCU: H8/3003

Function: DMAC (block transfer)

2.14.1 Specifications

- 6-byte data set in RAM is transferred to the I/O ports and output as 16-bit data each time an external signal falling edge is detected (see figure 2.76)
- DMAC startup is by external signal falling edge detection

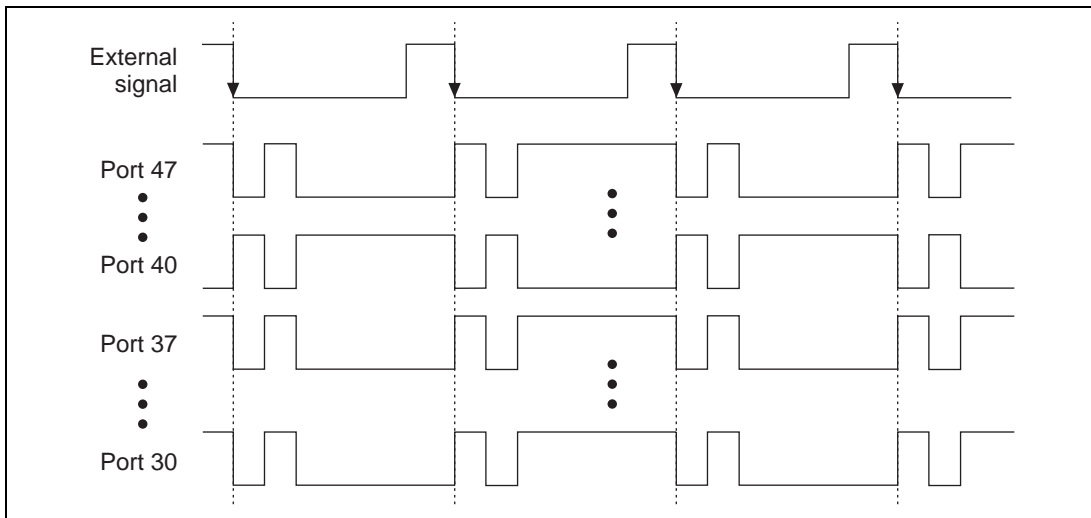


Figure 2.76 Example of Waveform Output

2.14.2 Description of Functions Used

In this sample task, Group 0 DMAC/ch0A and ch0B are used for block transfer.

Figure 2.77 shows a block diagram of the DMAC used in the sample task. The following DMAC functions are used in block transfer:

- DMAC startup due to external request (DMAC startup by DREQ)
- Data transfer in block units on DMAC startup (block transfer mode)
- Interrupt generation at end of transmit (DEND0 interrupt)

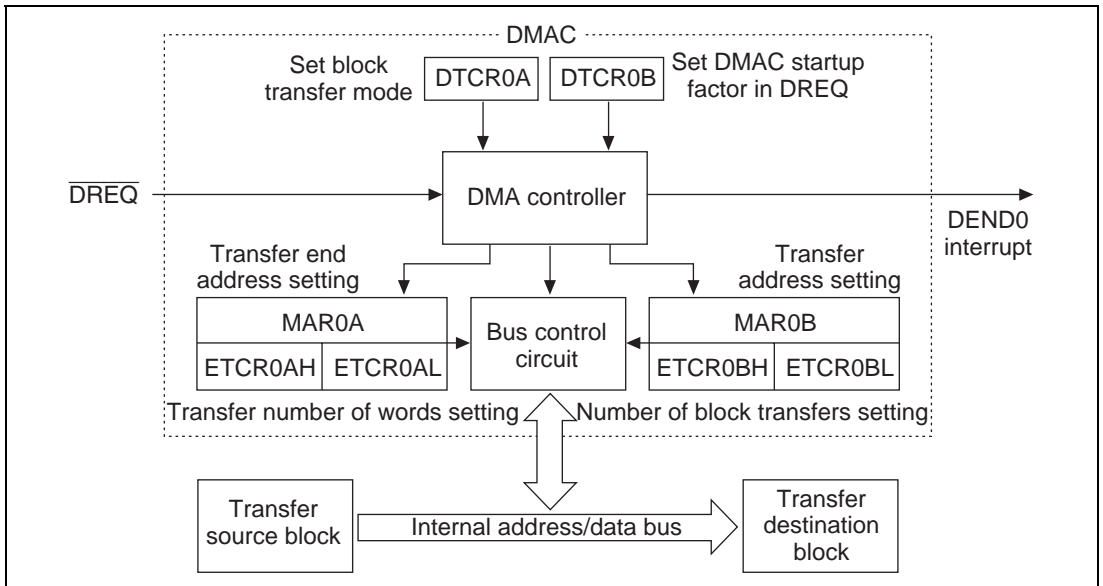


Figure 2.77 DMA Controller Block Diagram

Table 2.74 shows the DMAC function allocation for block transfer.

Table 2.74 DMAC Function Allocation

DMAC Function	Function
DREQ	Inputs external pulse acting as DMAC startup trigger
DTCR0A	Sets DMAC to block transfer mode
DTCR0B	Sets external pulse falling edge as DMAC startup trigger
MAR0A	Sets transfer source address
MAR0B	Sets transfer destination address
ETCR0AH, ETCR0AL	Sets transfer number of words
ETCR0BH, ETCR0BL	Sets number of transfers

2.14.3 Description of Operations

Figure 2.78 shows the principle of operation. Block transfer is conducted due to hardware and software processing with the timing shown in figure 2.78.

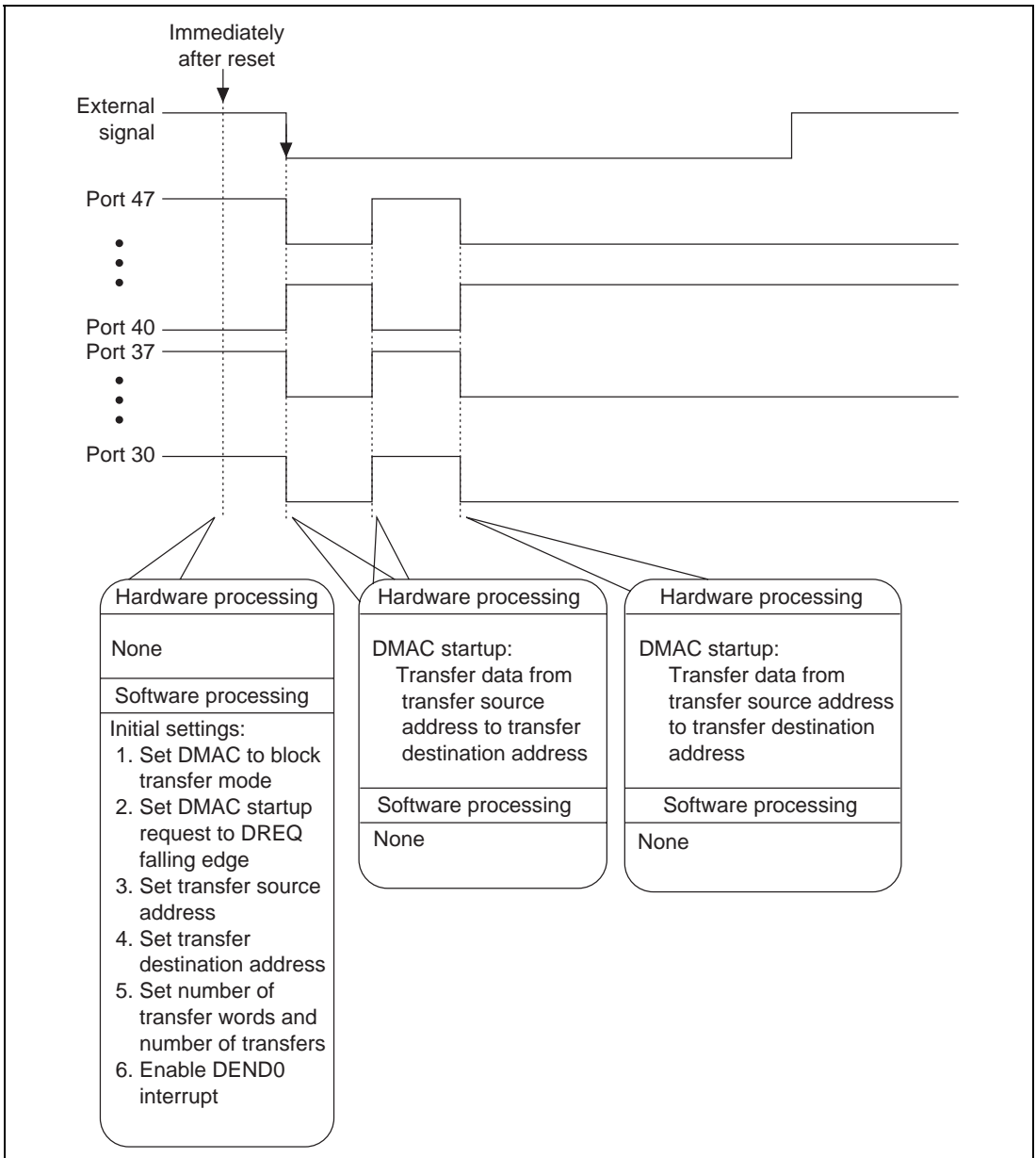


Figure 2.78 Principle of Block Transfer Operation

2.14.4 Description of Software

The software for block transfer is described below.

Table 2.75 Description of Modules

Module Name	Label Name	Function
Main routine	BLKMN	Initial DMAC setting
Block transfer end	BLKEND	Activated by TEND interrupt and re-setting DMAC

Description of Arguments: No arguments between modules with this sample task.

Table 2.76 Description of Internal Registers

Register Name	Function	Module Name
DTCR0A	Sets DMAC to block transfer mode	Main routine
DTCR0B	Sets DMAC startup trigger to external pulse falling edge	Main routine
MAR0A	Sets transfer source address	Main routine
MAR0B	Sets transfer destination address	Main routine
ETC0AH, ETC0AL	Sets number of transfer words	Main routine
ETC0BH	Sets number of transfers	Main routine

Table 2.77 Description of General Registers

Module Name	Register Name	Function
Main routine	R0L R0	Used as work register during DMAC setting

Description of RAM: RAM not used in this sample task.

2.14.5 Flowcharts

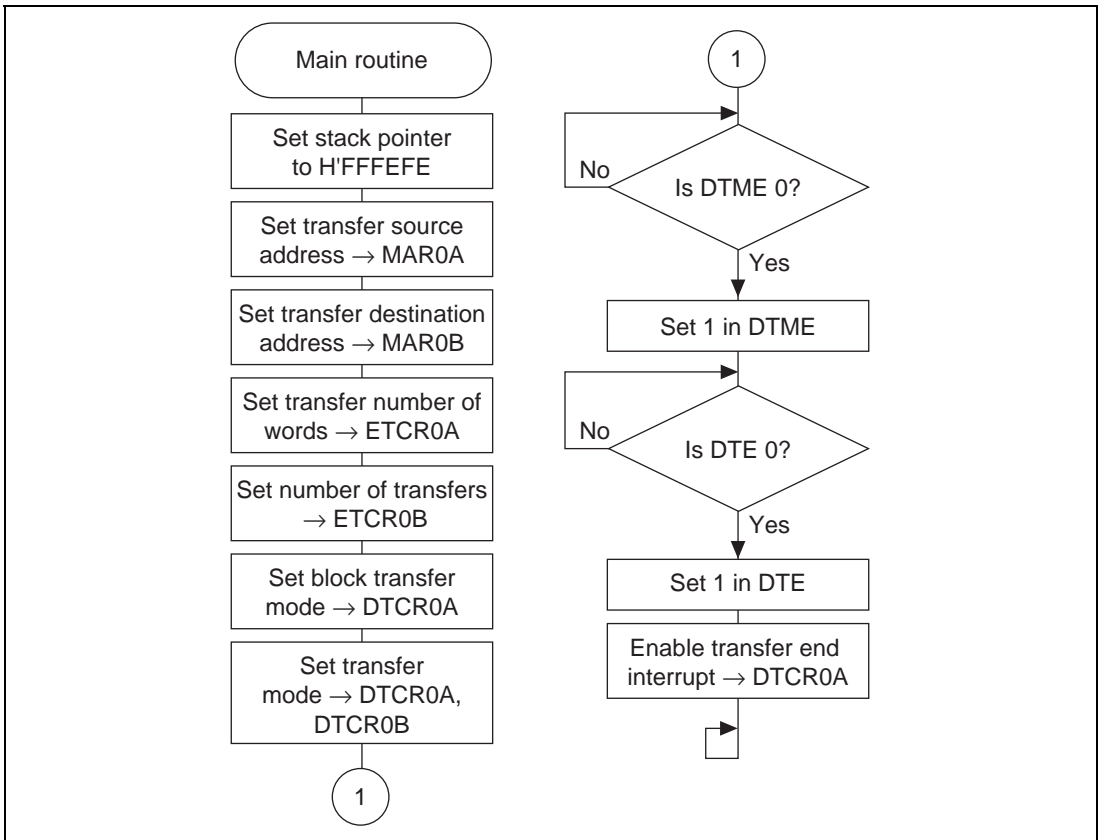


Figure 2.79 Main Routine Flowchart

2.14.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6
7          7          .CPU      300HA
8 000000    8          .SECTION  VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES      .DATA.L   BLKMN
11         11 ;
12 00001C    12          .ORG      H'00001C
13 00001C 00010000 13 NMI      .DATA.L   BLKMN
14         14 ;
15 000030    15          .ORG      H'000030
16 000030 00010000 16 IRQ0     .DATA.L   BLKMN
17 000034 00010000 17 IRQ1     .DATA.L   BLKMN
18 000038 00010000 18 IRQ2     .DATA.L   BLKMN
19 00003C 00010000 19 IRQ3     .DATA.L   BLKMN
20 000040 00010000 20 IRQ4     .DATA.L   BLKMN
21 000044 00010000 21 IRQ5     .DATA.L   BLKMN
22         22 ;
23 000050    23          .ORG      H'000050
24 000050 00010000 24 WOVI     .DATA.L   BLKMN
25 000054 00010000 25 CMI      .DATA.L   BLKMN
26         26 ;
27 000060    27          .ORG      H'000060
28 000060 00010000 28 TGAI0    .DATA.L   BLKMN
29 000064 00010000 29 TGBB0    .DATA.L   BLKMN
30 000068 00010000 30 TOVI0    .DATA.L   BLKMN
31         31 ;
32 000070    32          .ORG      H'000070
33 000070 00010000 33 TGAI1    .DATA.L   BLKMN
34 000074 00010000 34 TGBI1    .DATA.L   BLKMN
35 000078 00010000 35 TOVI1    .DATA.L   BLKMN
36         36 ;
37 000080    37          .ORG      H'000080
38 000080 00010000 38 TGAI2    .DATA.L   BLKMN
39 000084 00010000 39 TGBI2    .DATA.L   BLKMN
40 000088 00010000 40 TOVI2    .DATA.L   BLKMN
41         41 ;
42 000090    42          .ORG      H'000090
43 000090 00010000 43 TGAI3    .DATA.L   BLKMN

```

```

44 000094 00010000      44  TGBI3      .DATA.L  BLKMN
45 000098 00010000      45  TOVI3      .DATA.L  BLKMN
46                                     46  ;
47 0000A0                                     47          .ORG      H'0000A0
48 0000A0 00010000      48  TGAI4      .DATA.L  BLKMN
49 0000A4 00010000      49  TGBI4      .DATA.L  BLKMN
50 0000A8 00010000      50  TOVI4      .DATA.L  BLKMN
51                                     51  ;
52 0000B0                                     52          .ORG      H'0000B0
53 0000B0 00010000      53  DEND0     .DATA.L  BLKMN
54 0000B4 00010000      54  DEND1     .DATA.L  BLKMN
55 0000B8 00010000      55  DEND2     .DATA.L  BLKMN
56 0000BC 00010000      56  DEND3     .DATA.L  BLKMN
57                                     57  ;
58 0000D0                                     58          .ORG      H'0000D0
59 0000D0 00010000      59  ERI0      .DATA.L  BLKMN
60 0000D4 00010000      60  RXI0      .DATA.L  BLKMN
61 0000D8 00010000      61  TXI0      .DATA.L  BLKMN
62 0000DC 00010000      62  TEND0     .DATA.L  BLKMN
63 0000E0 00010000      63  ERI1      .DATA.L  BLKMN
64 0000E4 00010000      64  RXI1      .DATA.L  BLKMN
65 0000E8 00010000      65  TXI1      .DATA.L  BLKMN
66 0000EC 00010000      66  TEND1     .DATA.L  BLKMN
67 0000F0 00010000      67  ADI       .DATA.L  BLKMN
68                                     68  ;
69                                     69  ;*****
70                                     70  ;*                                     *
71                                     71  ;*          SYMBOL DEFINITIONS          *
72                                     72  ;*                                     *
73                                     73  ;*****
74                                     74  ;
75                                     75  ;***** DMAC0A *****
76          00FFFF20      76  MAR0AR:   .EQU      H'FFFF20   ;Memory address register
77          00FFFF24      77  ETCR0AH:  .EQU      H'FFFF24   ;Transmit cunt register 'H'
78          00FFFF25      78  ETCR0AL:  .EQU      H'FFFF25   ;Transmit cunt register 'L'
79          00FFFF27      79  DTCR0A:   .EQU      H'FFFF27   ;Data ransfer control
                                     ;register
80          00000007      80  DTE:      .EQU      7          ;Data transfer enable bit
81          00000003      81  DTIE     .EQU      3          ;Data transfer interrupt
                                     ;enable bit
82                                     82  ;***** DMAC0B *****
83          00FFFF28      83  MAR0BR:   .EQU      H'FFFF28   ;Memory address register
84          00FFFF2C      84  ETCR0BH:  .EQU      H'FFFF2C   ;Transmit cunt register 'H'
85                                     85  ;
86          00FFFF2F      86  DTCR0B   .EQU      H'FFFF2F   ;Data transfer control
                                     ;register

```

```

87          00000007          87 DTME      .EQU      7          ;Data transfer master
                                           ;enable bit

88          ;***** PORT 9 *****

89          00FFFFFFD0      89 P9DDR:    .EQU      H'FFFFFFD0      ;Port 9 data direction
                                           ;register

90          00FFFFFFD2      90 P9DR:     .EQU      H'FFFFFFD2      ;Port 9 data register

91          ;***** PORT A *****

92          00FFFFFFD1      92 PADDR:    .EQU      H'FFFFFFD1      ;Port A data direction
                                           ;register

93          00FFFFFFD3      93 PADR:     .EQU      H'FFFFFFD3      ;Port A data register

94          94 ;

95          95 ;*****

96          96 ;*

97          97 ;*      MAIN PROGRAM : BLKMN

98          98 ;*

99          99 ;*****

100         100 ;

101         010000          101          .SECTION  PROG,CODE,LOCATE=H'010000

102         102 ;

103         00010000      103 BLKMN:    .EQU      $

104         010000      7A0700FFFEFE      104          MOV.L     #H'FFFEFE,SP ;Set stack pointer

105         010006      F8FF          105          MOV.B     #H'FF,R0L

106         010008      38D0          106          MOV.B     R0L,@P9DDR ;Port 9 is output port

107         01000A      38D1          107          MOV.B     R0L,@PADDR ;Port A is output port

108         108 ;

109         01000C      7A0000010054      109          MOV.L     #SRCADD,ER0 ;Set source address

110         010012      01006B80FF20      110          MOV.L     ER0,@MAR0AR ;

111         010018      7A0000FFFFFFD2      111          MOV.L     #P9DR,ER0 ;Set distnation address

112         01001E      01006B80FF28      112          MOV.L     ER0,@MAR0BR ;

113         113 ;

114         010024      790000002      114          MOV.W     #2,R0 ;Set transfer count=2

115         010028      6B80FF2C          115          MOV.W     R0,@ETCR0BH

116         116 ;

117         01002C      F803          117          MOV.B     #H'03,R0L ;Set block size = 3word

118         01002E      3824          118          MOV.B     R0L,@ETCR0AH

119         010030      3825          119          MOV.B     R0L,@ETCR0AL

120         120 ;

121         010032      F80E          121          MOV.B     #B'00001110,R0L ;Distnatino address
fix

122         010034      382F          122          MOV.B     R0L,@DTCR0B ;DREQ falling edge
                                           ;request

123         123 ;

124         010036      F857          124          MOV.B     #B'01010111,R0L ;Block trans.mode,word
                                           ;size

125         010038      3827          125          MOV.B     R0L,@DTCR0A ;Source address increment

126         126 ;

```



```

127 01003A          127 BLKMN01:
128 01003A 7E2F7370 128          BTST      #DTME,@DTCR0B ;Is DTME 'Low' ?
129 01003E 46FA     129          BNE      BLKMN01      ;No
130                130 ;
131 010040 7F2F7070 131          BSET      #DTME,@DTCR0B ;Set DTME
132                132 ;
133 010044          133 BLKMN02:
134 010044 7E277370 134          BTST      #DTE,@DTCR0A ;DTE is 'Low' ?
135 010048 46FA     135          BNE      BLKMN02      ;No
136                136 ;
137 01004A 7F277070 137          BSET      #DTE,@DTCR0A ;Set DTE
138 01004E 7F277030 138          BSET      #DTIE,@DTCR0A ;Enable DEND0 interrupt
139                139 ;
140 010052          140 BLKMN99:
141 010052 40FE     141          BRA      BLKMN99
142                142 ;
143                143 ;*****
144                144 ;*                                     *
145                145 ;*          DATA TABLE (OUTPUT PATTERN)          *
146                146 ;*                                     *
147                147 ;*****
148          00010054 148 SRCADD:  .EQU      $
149 010054 1111     149          .DATA.W   H'1111
150 010056 2222     150          .DATA.W   H'2222
151 010058 3333     151          .DATA.W   H'3333
152 01005A 4444     152          .DATA.W   H'4444
153                153 ;
154                154          .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```


3.1 High-Speed Data Output

MCU: H8/3003

Function: ITU, TPC, DMAC

3.1.1 Specifications

- Output of 12-bit data at every rising edge detection of external signal (see figure 3.1)

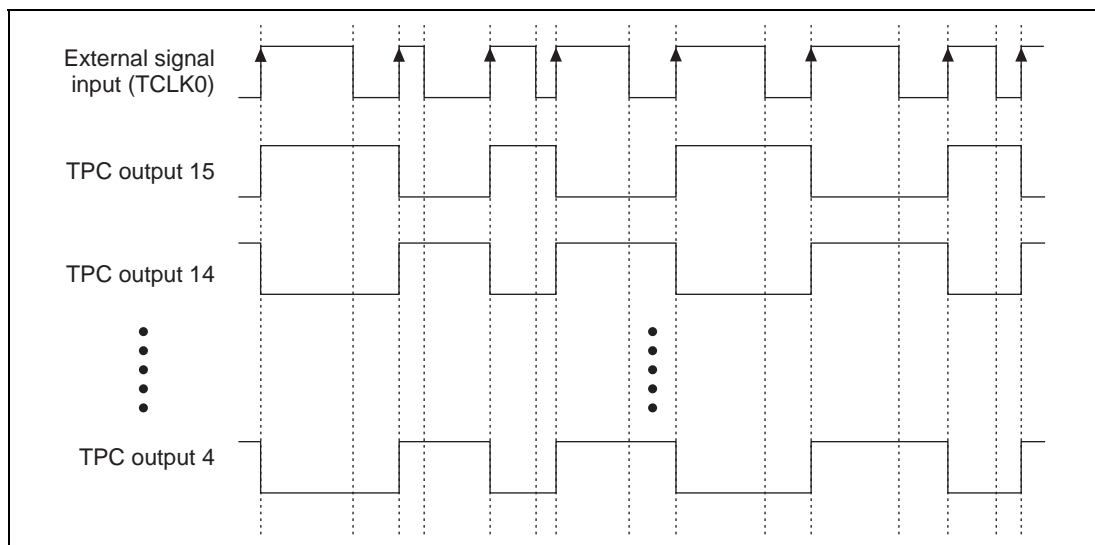


Figure 3.1 Example of 12-Bit Data Output

3.1.2 Method

Figure 3.2 shows a block diagram of on-chip functions used in this sample task. The ITU generates a compare/match signal to startup the DMAC each time an external signal is input. The DMAC transfers output pattern data table values to the TPC.

Repeating the above operations achieves realtime port control in response to the external signal's rising edge.

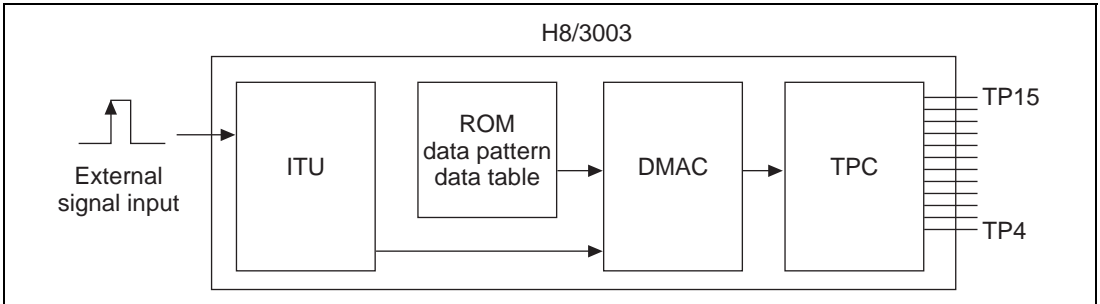


Figure 3.2 H8/3003 Block Diagram

3.1.3 Description of Functions Used

Figure 3.3 shows a block diagram of the onchip functions used in this sample task. The following H8/3003 functions are used for high-speed data output.

Output Pattern Data Table: Sets data pattern output from TPC in ROM.

ITU: Clears timer counter with each compare/match A. Also starts up DMAC0 and TPC. (H'0001 set in GRA and counting conducted by rising edge of external clock).

DMAC0A: Starts up by ITU compare/match A, and transfers output data from output pattern data table to NDR.

TPC: Starts up by ITU compare/match A, and uses DMAC to output data without CPU intervention. 12-bit data is output in this task, although a maximum of 16-bit data output is possible.

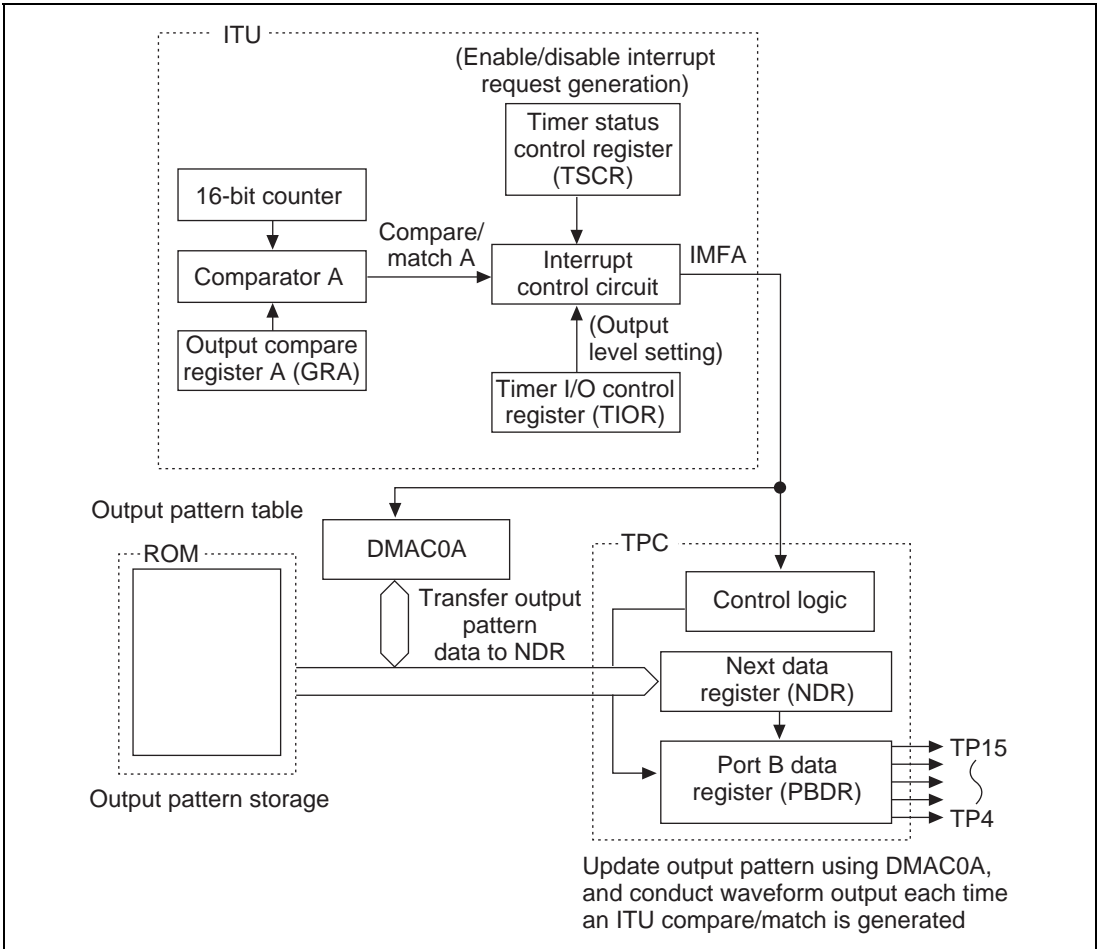


Figure 3.3 High-Speed Data Output Block Diagram

Table 3.1 shows the function allocation for this sample task. On-chip I/O functions are allocated for high-speed data output.

Table 3.1 H8/3003 Function Allocation

H8/3003 Function	Function	
ITU (ch0)	TCLK0	External signal rising edge detection
	TCNT0	16 bit counter
	GRA0	Output compare register
	TCR0	Counter clock select and counter clear factor select
	TIOR0	Sets GRA to output compare register
	TSR0	Indicates compare/match and overflow status
	TIER0	Selects interrupt enable/disable
	TSTR	Sets enable/disable TCNT counting operation
DMAC0A	DTCR0A	Controls DMAC operation
	MAR0A	Sets start address of output pattern data table
	IORA0A	Sets NDR address
	ETCR0A	Sets number of transfers
TPC	PADDR	Sets TPC output pin
	PADR	Stores TPC output data TP7 – TP4
	PBDDR	Sets TPC output pin
	PBDR	Stores TPC output data TP15 – TP8
	TPMR	Selects TPC non-overlap mode
	TPCR	Selects output trigger signal for TPC output
	NDERB	Sets enable/disable for TPC output TP15 – TP8
	NDBRA	Sets enable/disable for TPC output TP7 – TP4
	NDRB	Stores next output data of TPC output TP15 – TP8
	NDRA	Stores next output data of TPC output TP7 – TP4
	TP15 – TP4	TPC data output pin

3.1.4 Description of Operations

Figure 3.4 shows the principle of operation. High-speed data output is achieved through H8/3003 hardware and software processing.

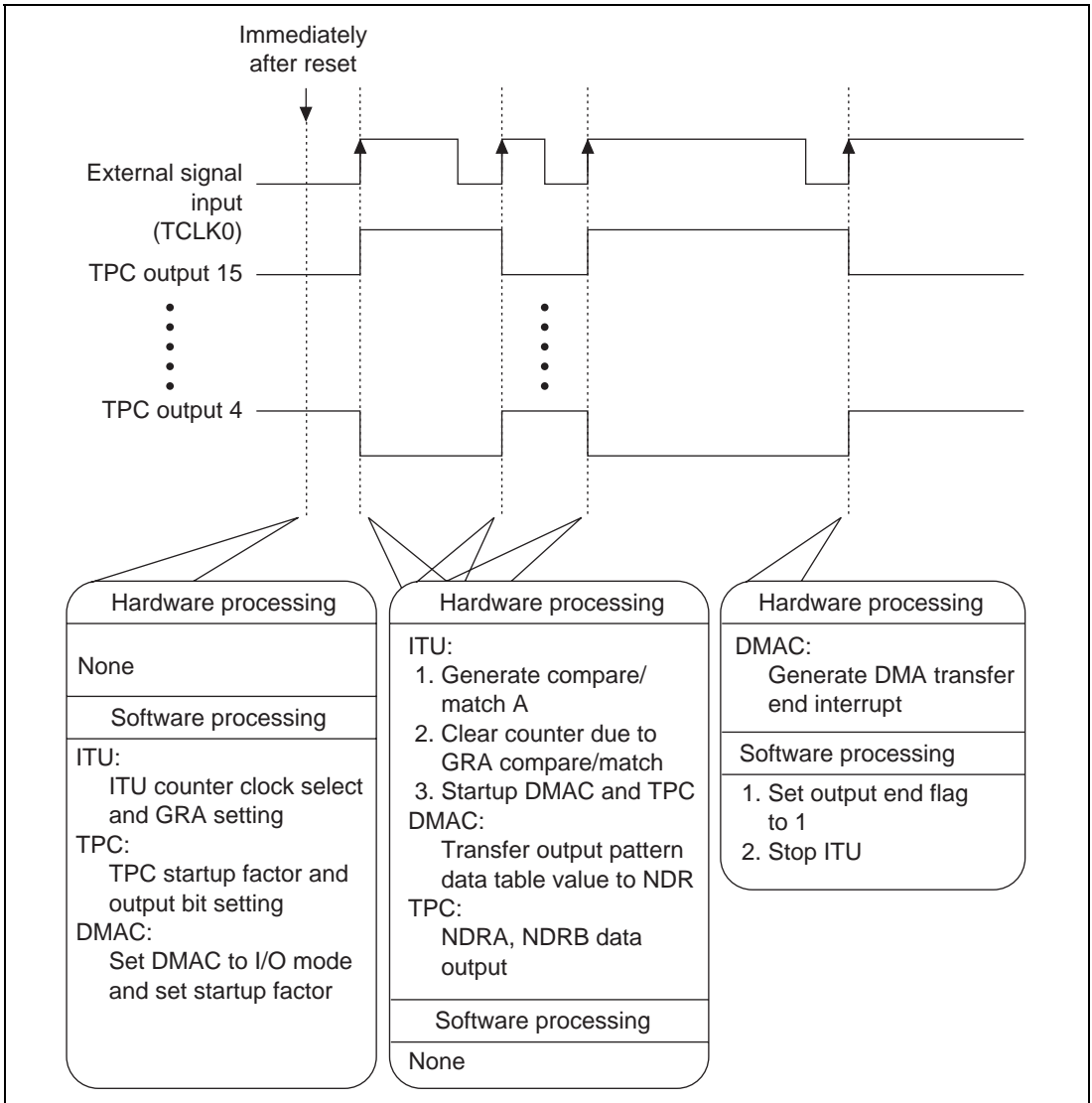


Figure 3.4 Principle of High-Speed Data Output Operation

3.1.5 Description of Software

The software for high-speed data output is described below.

Table 3.2 Description of Modules

Module Name	Label Name	Function
Main routine	STPCMn	ITU, TPC and DMAC0A initial settings
DMA interrupt	DMAEND	Sets output end flag

Table 3.3 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
DMA_FL A	Flag indicating that output has completely ended. 0: output of all bits not ended, 1: output of all bits ended	1 bit	Main routine	I/O
			DMA interrupt	I/O

Table 3.4 Description of Internal Registers

Onchip Function	Register Name	Function
ITU	GRA	Sets output compare value (H'01)
	TCR	Sets ITU as follows: <ul style="list-style-type: none">• Counter clear with GRA compare/match• Count with external signal rising edge• Count at TCLKA pin
	TIOR	Sets GRA to output compare register and prohibits pin output
	TIER	Enables IMFA interrupts
	TSTR	Starts TCNT0 counting operation

Onchip Function	Register Name	Function	
TPC	PADDR	Enables TPC output from TP7 – TP4	
	PBDDR	Enables TPC output from TP15 – TP8	
	PADR	Stores output data from TP7 – TP4	
	PBDR	Stores output pattern from TP15 – TP8	
	TPMR	Sets normal operation from TP15 – TP4	
	TPCR	Sets TP15 – TP4 output trigger to ITU0 compare/match	
	NDERA	Enables TPC output from TP7 – TP4	
	NDERB	Enables TPC output from TP15 – TP8	
	NDRA	Stores next output pattern data	
	NDRB	Stores next output pattern data	
DMAC0A	DTCR0A	Sets DMAC0A as follows: <ul style="list-style-type: none"> • Data size to word size • MAR to increment • Data transfer to I/O mode • Startup factor to ITU ch0 compare/match A 	
		MAR0A	Sets transfer source address of output pattern data table
		IOAR0A	Sets NDRB address (transfer destination)
		ETCR0A	Sets number of transfers

Description of RAM: RAM not used in this sample task.

Table 3.5 Description of Data Table

Table Name	Function	Data Length	Data Volume
OPAT_TAB	Stores data output from TPC	1 word	15 words

Description of General Registers: Used as work register during data setting.

3.1.6 Flowcharts

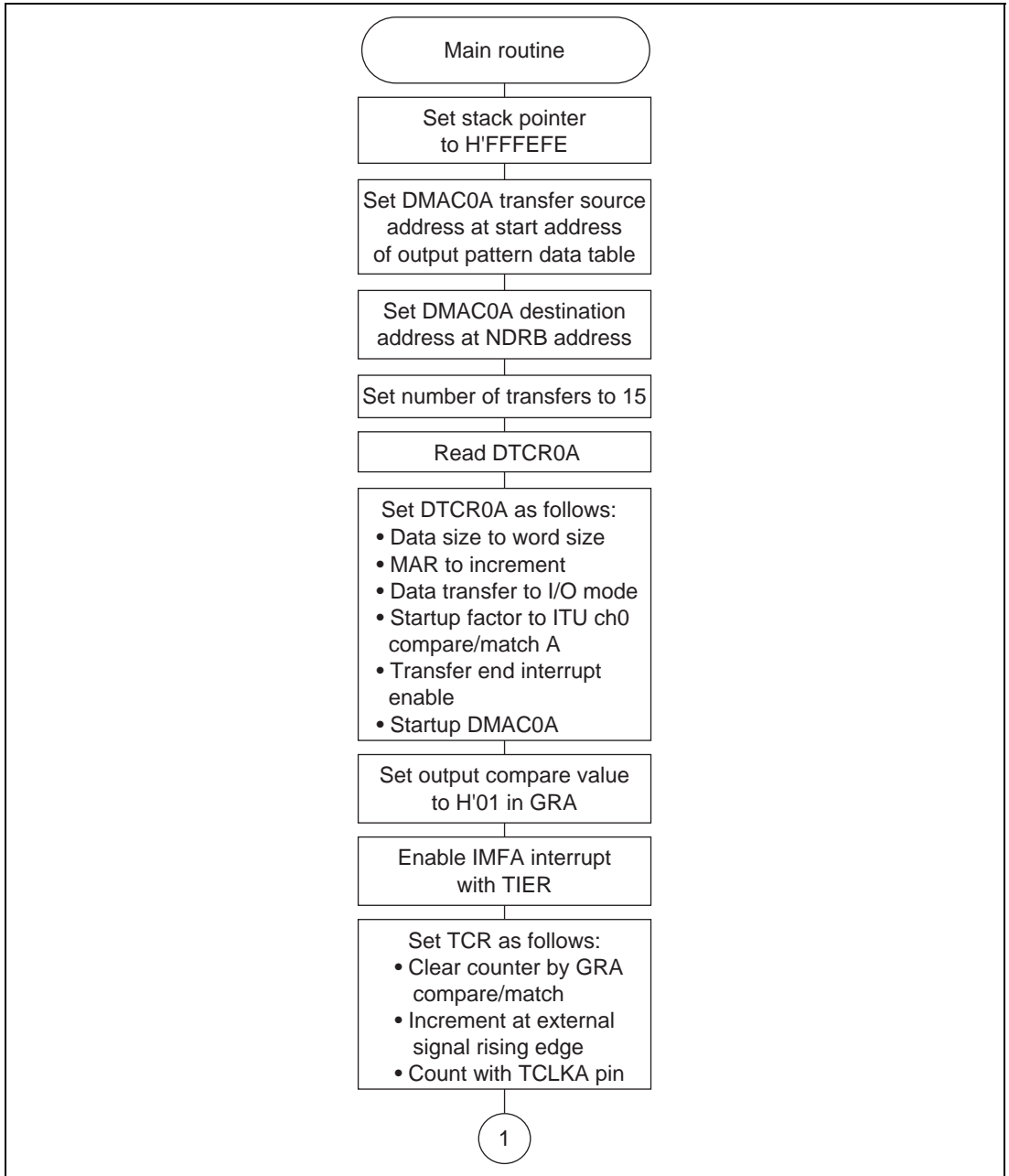


Figure 3.5 Main Routine Flowchart (1)

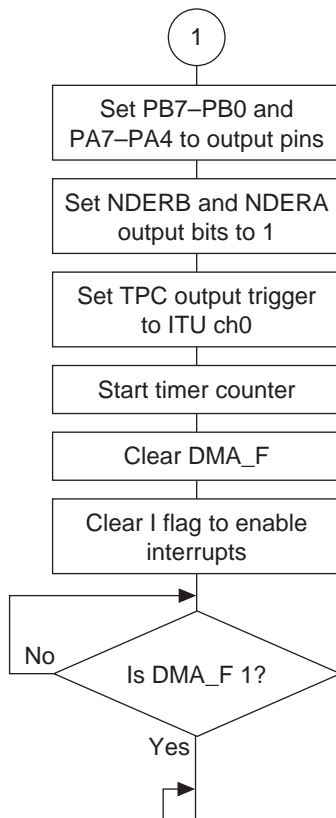


Figure 3.5 Main Routine Flowchart (2)

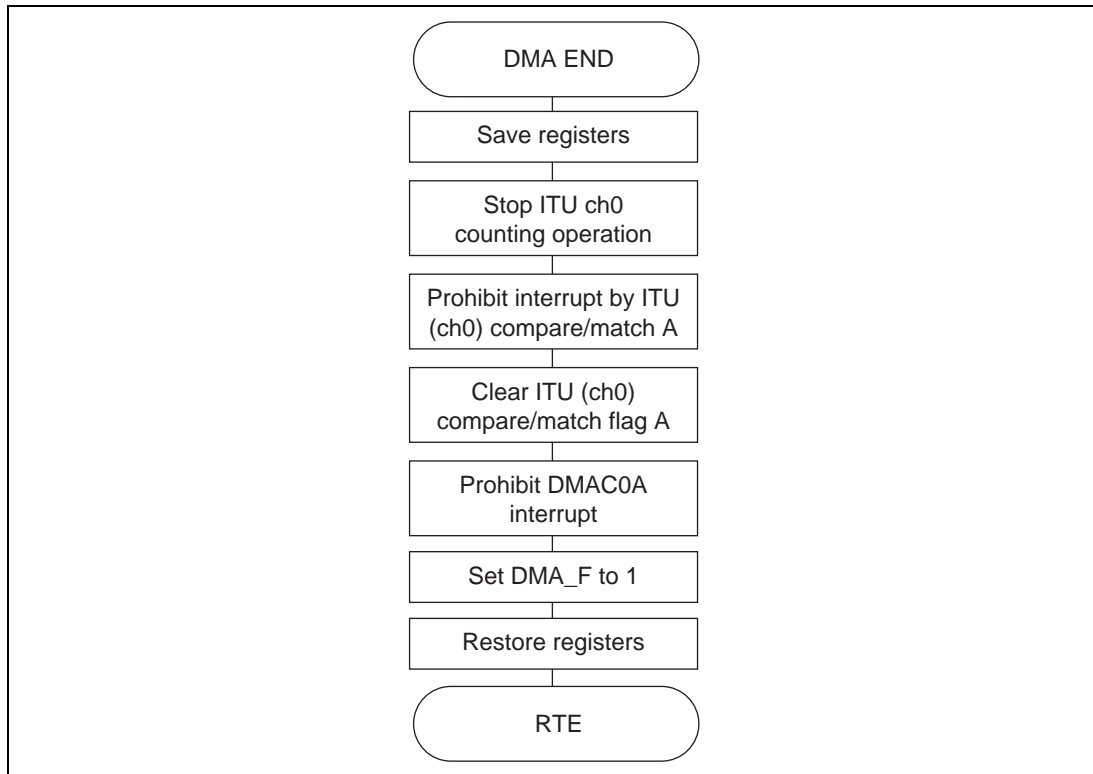


Figure 3.6 DMA END Flowchart

3.1.7 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6
7          7          .CPU          300HA
8 000000    8          .SECTION    VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000   00010000 10 RES          .DATA.L    STPCMN
11          11 ;
12 00001C   00010000 12          .ORG          H'00001C
13 00001C   00010000 13 NMI          .DATA.L    STPCMN
14          14 ;
15 000030   00010000 15          .ORG          H'000030
16 000030   00010000 16 IRQ0         .DATA.L    STPCMN
17 000034   00010000 17 IRQ1         .DATA.L    STPCMN
18 000038   00010000 18 IRQ2         .DATA.L    STPCMN
19 00003C   00010000 19 IRQ3         .DATA.L    STPCMN
20 000040   00010000 20 IRQ4         .DATA.L    STPCMN
21 000044   00010000 21 IRQ5         .DATA.L    STPCMN
22 000048   00010000 22 IRQ6         .DATA.L    STPCMN
23 00004C   00010000 23 IRQ7         .DATA.L    STPCMN
24 000050   00010000 24 WOVI0        .DATA.L    STPCMN
25 000054   00010000 25 CMI          .DATA.L    STPCMN
26          26 ;
27 000060   00010000 27          .ORG          H'000060
28 000060   00010000 28 IMIA0        .DATA.L    STPCMN
29 000064   00010000 29 IMIB0        .DATA.L    STPCMN
30 000068   00010000 30 OVI0         .DATA.L    STPCMN
31          31 ;
32 000070   00010000 32          .ORG          H'000070
33 000070   00010000 33 IMIA1        .DATA.L    STPCMN
34 000074   00010000 34 IMIB1        .DATA.L    STPCMN
35 000078   00010000 35 OVI1         .DATA.L    STPCMN
36          36 ;
37 000080   00010000 37          .ORG          H'000080
38 000080   00010000 38 IMIA2        .DATA.L    STPCMN
39 000084   00010000 39 IMIB2        .DATA.L    STPCMN
40 000088   00010000 40 OVI2         .DATA.L    STPCMN
41          41 ;
42 000090   00010000 42          .ORG          H'000090
43 000090   00010000 43 IMIA3        .DATA.L    STPCMN

```

```

44 000094 00010000      44 IMIB3      .DATA.L      STPCMN
45 000098 00010000      45 OVI3      .DATA.L      STPCMN
46                               46 ;
47 0000A0                               47          .ORG          H'0000A0
48 0000A0 00010000      48 IMIA4      .DATA.L      STPCMN
49 0000A4 00010000      49 IMIB4      .DATA.L      STPCMN
50 0000A8 00010000      50 OVI4      .DATA.L      STPCMN
51                               51 ;
52 0000B0                               52          .ORG          H'0000B0
53 0000B0 0001005C      53 DEND0A     .DATA.L      DMAEND
54 0000B4 00010000      54 DEND0B     .DATA.L      STPCMN
55 0000B8 00010000      55 DEND1A     .DATA.L      STPCMN
56 0000BC 00010000      56 DEND1B     .DATA.L      STPCMN
57 0000C0 00010000      57 DEND2A     .DATA.L      STPCMN
58 0000C4 00010000      58 DEND2B     .DATA.L      STPCMN
59 0000C8 00010000      59 DEND3A     .DATA.L      STPCMN
60 0000CC 00010000      60 DEND3B     .DATA.L      STPCMN
61 0000D0 00010000      61 ERI0      .DATA.L      STPCMN
62 0000D4 00010000      62 RXI0      .DATA.L      STPCMN
63 0000D8 00010000      63 TXI0      .DATA.L      STPCMN
64 0000DC 00010000      64 TEI0      .DATA.L      STPCMN
65 0000E0 00010000      65 ERI1      .DATA.L      STPCMN
66 0000E4 00010000      66 RXI1      .DATA.L      STPCMN
67 0000E8 00010000      67 TXI1      .DATA.L      STPCMN
68 0000EC 00010000      68 TEI1      .DATA.L      STPCMN
69 0000F0 00010000      69 ADI       .DATA.L      STPCMN
70                               70 ;
71                               71 ;*****
72                               72 ;*                                     *
73                               73 ;*          RAM ALLOCATION                *
74                               74 ;*                                     *
75                               75 ;*****
76                               76 ;
77 FFFF00                               77          .SECTION   RAM,DATA,LOCATE=H'FFFF00
78                               78 ;
79 FFFF00 00000001      79 DMA_FL A: .RES.B      1          ;Flag area
80          00000000      80 DMA_END F: .EQU       0          ;DMA end flag
81                               81 ;
82                               82 ;*****
83                               83 ;*                                     *
84                               84 ;*          SYMBOL DEFINITIONS            *
85                               85 ;*                                     *
86                               86 ;*****
87                               87 ;
88                               88 ;***** DMAC0A *****
89          00FFFF20      89 MAR0AR: .EQU       H'FFFF20      ;Memory address register

```

```

90      00FFFF24      90  ETCR0AH: .EQU      H'FFFF24      ;Transmit cunt register
91      00FFFF26      91  IOAR0A: .EQU      H'FFFF26      ;I/O address register
92      00FFFF27      92  DTCR0A: .EQU      H'FFFF27      ;Data transfer control
;register
93      00000007      93  DTE:      .EQU      7              ;Data transfer enable bit
94      ;
95      ;***** ITU *****
96      00FFFF60      96  TSTR:      .EQU      H'FFFF60      ;Timer start register
97      ;
98      ;***** ITU ch0 *****
99      00FFFF64      99  TCR0:      .EQU      H'FFFF64      ;timer control register
100     00FFFF65      100 TIOR0:      .EQU      H'FFFF65      ;timer I/O control register
101     00FFFF66      101 TIER0:      .EQU      H'FFFF66      ;timer interrupt enable
;register
102     00FFFF67      102 TSR0       .EQU      H'FFFF67      ;timer status register
103     00FFFF6A      103 GRA0H:      .EQU      H'FFFF6A      ;General register
104     ;
105     ;***** TPC *****
106     00FFFA0      106 TPMR:      .EQU      H'FFFA0      ;TPC output mod register
107     00FFFA1      107 TPCR:      .EQU      H'FFFA1      ;TPC output control register
108     00FFFA2      108 NDERB:      .EQU      H'FFFA2      ;Next data enable register B
109     00FFFA3      109 NDERA:      .EQU      H'FFFA3      ;Next data enable register A
110     00FFFD1      110 PADDR:      .EQU      H'FFFD1      ;Port A data direction
;register
111     00FFFD4      111 PBDDR:      .EQU      H'FFFD4      ;Port B data direction
;register
112     ;
113     ;*****
114     ;*
115     ;*      MAIN PROGRAM : STPCMN
116     ;*
117     ;*****
118     ;
119  010000      119      .SECTION   PROG, CODE, LOCATE=H'010000
120     ;
121     00010000      121 STPCMN:      .EQU      $
122  010000  7A0700FFFEFE      122      MOV.L      #H'FFFEFE, SP      ;Init. stack pointer
123     ;
124     ;***** INITIALZE ITU *****
125  010006  79000000      125      MOV.W      #H'0, R0
126  01000A  6B80FF6A      126      MOV.W      R0, @GRA0H      ;Set non overlap time
127  01000E  F801      127      MOV.B      #H'01, R0L
128  010010  3866      128      MOV.B      R0L, @TIER0      ;Init. TIER0(1->IMIEA)
129  010012  F824      129      MOV.B      #H'24, R0L
130  010014  3864      130      MOV.B      R0L, @TCR0      ;Init. TCR0
; (1->CCLR0, TPSC2)

```

```

131          131 ;
132          132 ;***** INITIALZE DMAC0A *****
133 010016 7A0000010076 133          MOV.L      #OPAT_TAB,ER0
134 01001C 01006B80FF20 134          MOV.L      ER0,@MAR0AR      ;Set MAR0A
135 010022 F8A4          135          MOV.B      #H'A4,R0L
136 010024 3826          136          MOV.B      R0L,@IOAR0A      ;Set IOAR0A
137 010026 7900000F 137          MOV.W      #H'0F,R0
138 01002A 6B80FF24 138          MOV.W      R0,@ETCR0AH      ;Store transfer count
139 01002E 2827          139          MOV.B      @DTCR0A,R0L      ;Read DTCR0A
140 010030 F848          140          MOV.B      #H'48,R0L
141 010032 3827          141          MOV.B      R0L,@DTCR0A      ;Init. DTCR0A
142 010034 7F277070 142          BSET       #DTE,@DTCR0A      ;Start DMAC0A
143          143 ;
144          144 ;***** INITIALZE TPC *****
145 010038 F8F0          145          MOV.B      #H'F0,R0L
146 01003A 38D1          146          MOV.B      R0L,@PADDR      ;Init. PADDR
147 01003C F8FF          147          MOV.B      #H'FF,R0L
148 01003E 38D4          148          MOV.B      R0L,@PBDDR      ;Init. PBDDR
149 010040 38A2          149          MOV.B      R0L,@NDERB      ;Set output TPC
150 010042 F8F0          150          MOV.B      #H'F0,R0L
151 010044 38A3          151          MOV.B      R0L,@NDERA      ;Set output TPC
152 010046 F800          152          MOV.B      #H'00,R0L
153 010048 38A1          153          MOV.B      R0L,@TPCR      ;Init. TPCR
154          154 ;
155 01004A F801          155          MOV.B      #H'01,R0L
156 01004C 3860          156          MOV.B      R0L,@TSTR      ;Start ITU ch0
157 01004E 7F007200 157          BCLR      #DMA_ENDF,@DMA_FL A ;Clear DMA_ENDF
158 010052 0700          158          LDC.B      #0,CCR          ;Enable interrupt
159          159 ;
160 010054 7E007300 160 STPCMN01: BTST      #DMA_ENDF,@DMA_FL ;DMA_ENDF = "1" ?
161 010058 47FA          161          BEQ       STPCMN01      ;No
162 01005A 40FE          162 STPCMN99: BRA      STPCMN99      ;Loop
163          163 ;
164          164 ;*****
165          165 ;*
166          166 ;*      NAME : DMAEND (End of DMAC0A transmit) *
167          167 ;*
168          168 ;*****
169          169 ;*
170          170 ;*      ENTRY : NOTHING *
171          171 ;*      RETURN : DMA_ENDF (DMAC end flag) *
172          172 ;*
173          173 ;*****
174          174 ;
175          175 DMAEND: .EQU      $
176 01005C 6DF0          176          PUSH.W   R0

```



```

177 01005E 7F607200      177      BCLR.B      #0,@TSTR      ;Stop ITU ch0
178 010062 7F667200      178      BCLR.B      #0,@TIER0     ;Disable timer
                                           ;comparemech interrupt
179 010066 7F677200      179      BCLR.B      #0,@TSR0     ;Clear timer
                                           ;comparemech flag
180 01006A 7F277230      180      BCLR.B      #3,@DTCR0A   ;Stop DMAC0A
181 01006E 7F007000      181      BSET        #DMA_ENDF,@DMA_FLA ;Set DMA_ENDF
182 010072 6D70          182      POP.W      R0
183 010074 5670          183      RTE
184          184      ;
185          185      ;*****
186          186      ;*                               *
187          187      ;*          DATA TABLE (OUTPUT PATTERN)          *
188          188      ;*                               *
189          189      ;*****
190 010076 1110222033304440 190 OPAT_TAB  .DATA.W      H'1110,H'2220,H'3330,H'4440,H'5550,
191 01007E 5550666077708880 191 *          H'6660,H'7770,H'8880,H'9990,H'AAA0,
192 010086 9990AAA0BBB0CCC0 192 *          H'BBB0,H'CCC0,H'DDD0,H'EEE0,H'FFF0
      01008E DDD0EEE0FFF0
193          193      ;
194          194      .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```

3.2 SCI Continuous Transmit/Receive

MCU: H8/3003

Function: ITU, DMAC

3.2.1 Specifications

- Setting of H8/3003 SCI to clock sync mode and continuous transmit/receive of 100 byte data between H8/3003 devices
- Using DMAC, setting for data transfer from memory to TDR and RDR to memory without CPU intervention
- The transmit side is the clock master

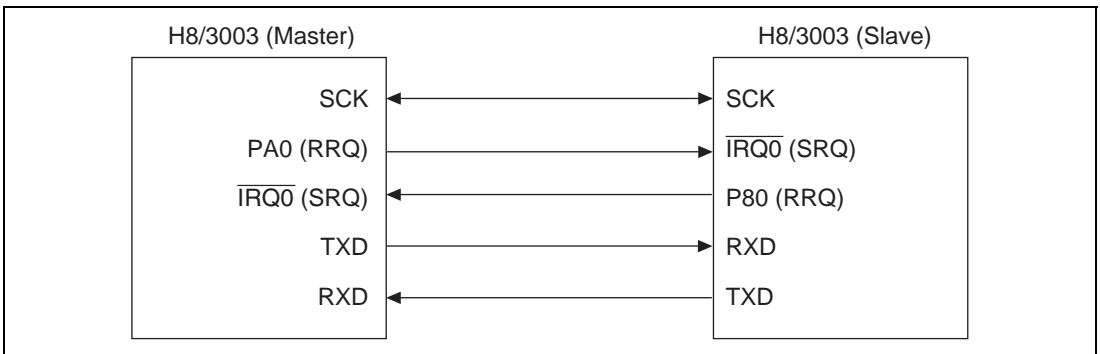


Figure 3.7 Block Diagram of Clock Sync SCI Interface by H8/3003

3.2.2 Method

The approach shown below has the I/O port control the signal line.

RRQ Signal (receive request): Receive request sent to slave side. After sending the receive request, 100 us later a serial clock is sent from SCK and serial data from TXD to the slave side.

SRQ Signal (send request): Request for transmission from slave side. After receiving the SRQ, 100 us later a serial clock is received from SCK and serial data from RXD is received from the slave side.

3.2.3 Description of Functions Used

Figure 3.8 shows the H8/3003 on-chip functions used in this sample task. The DMAC and SCI are used for high-speed serial communications.

Transmit Data Table: Sets transmit data in ROM

Receive Data Table: Sets receive data in RAM

DMAC0A:

- During transmit: Activated by transmit end interrupt, and transfers SCI data table contents to SCI
- During receive: Activated by receive end interrupt, and transfers receive data to SCI data table

SCI: Conducts serial data transmit and receive with H8/3003.

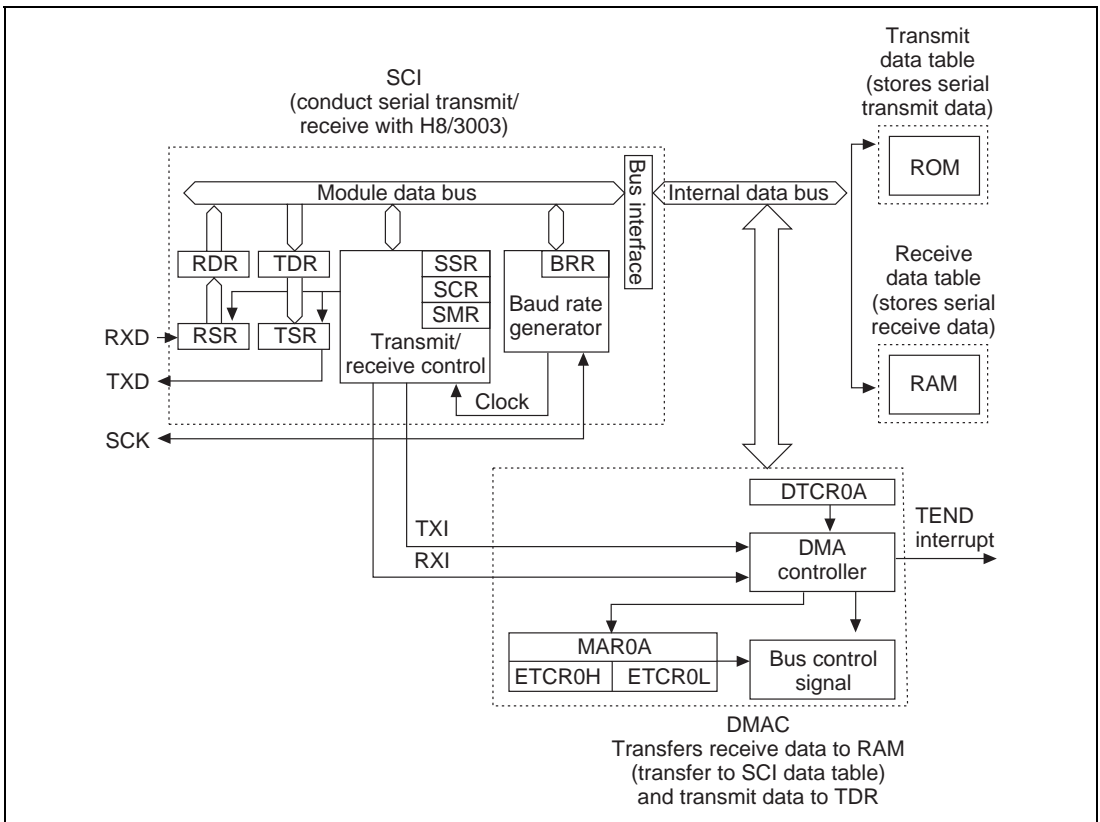


Figure 3.8 SCI and DMAC Block Diagram

Table 3.6 shows the function allocation with this sample task. SCI and DMAC functions are allocated for interfacing with H8/3003.

Table 3.6 H8/3003 Function Allocation

H8/3003 Function	Function	
SCI	SCK	Transmits transfer clock, and receives transfer clock during receive
	RXD	Receive data input pin
	TXD	Transmit data output pin
	SMR	Sets SCI in clock sync mode
	SCR	Enables receive interrupt, sets SCI to transmit mode
	SSR	Starts transmit
	RDR	Sets data received from H8/3003
	TDR	Sets data for transmission to H8/3003
	BRR	Sets the transfer rate
Port A	PADDR	Sets PORT A I/O
	PADR	Transmits RRQ
DMAC0A	DTCR0A	Controls each of the DMAC channel operations
	MAR0A	Sets transfer source address during startup by SCI transmit end interrupt.
		Sets transfer destination address during startup by SCI receive end interrupt
	IOAR0A	Sets transfer destination address during startup by SCI transmit end interrupt.
		Sets transfer source address during startup by SCI receive end interrupt
ETCR0A	Sets number of transfers	

3.2.4 Description of Software

The software for SCI continuous transmit/receive is described below.

Table 3.7 Description of Modules

Module Name	Label Name	Function
Main routine	HSCIMN	Initial setting of I/O port, SCI and DMAC
Data transmit	HSCITX	Sets transmit data in TDR and starts transmit
Data receive	HSCIRX	Activated by IRQ0 interrupt, and sets SCI to receive operation
DMA interrupt	DMAEND	Conducts WRIT_F and READ_F, and sets transmit end and receive end flags

Table 3.8 Description of Internal Registers

Onchip Function	Register Name	Function
SCI	SMR	Sets SCI as follows: <ul style="list-style-type: none"> • Sets SCI to clock sync mode • Sets baud rate generator's clock source to ϕ
	SCR	Sets SCI as follows during transmit and receive. Transmit operation: <ul style="list-style-type: none"> • Enable transmit data empty interrupt • Enable transmit operation • SCK pin to sync clock output Receive operation: <ul style="list-style-type: none"> • Enable receive data full interrupt • Enable receive operation • SCK pin to sync clock input
	SSR	Transmit: Clears TDRE and starts transmit operation Receive: Clears RDRF and enables receive operation
	RDR	Stores data received from H8/300H
	TDR	Sets data for transmission to H8/300H
	BRR	Sets transfer rate
	DMAC	DTCR0A
MAR0A		During transmit: Sets transfer source address of SCI data table During receive: Sets transfer destination address of SCI data table
IOAR0A		During transmit: Sets TDR (transfer destination) address During receive: Sets RDR (transfer source) address
ETCR0A		Sets number of transfers
I/O		PADDR
	PADR	Transmits RRQ

Onchip Function	Register Name	Function
Interrupt controller	IER	Enables IRQ0 interrupt
	ISR	Indicates IRQ0 input status

Table 3.9 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
R0L	Stores data for transmission	1 byte	Main routine	Input
			Data transmit	Output
WRIT_F	Flag indicating write action	1 bit	Data transmit	Input
			Data receive	Output
READ_F	Flag indicating read action	1 bit	Data transmit	Input
			Data receive	Output
WRIT_ERRF	Flag indicating transmit ended normally 0: Transmit normal end 1: Transmit end error	1 bit	Data transmit	Input
			Data receive	Output
READ_ERRF	Flag indicating receive ended normally 0: Receive normal end 1: Receive abnormal end	1 bit	Data transmit	Input
			Data receive	Output

Table 3.10 Description of RAM

Register Name	Function	Data Length	Module Name
READ_END	Flag indicating end of read operation	1 bit	DMA interrupt
WRIT_END	Flag indicating end of write operation	1 bit	DMA interrupt

Table 3.11 Description of Data Tables

Register Name	Function	Data Length	Data Volume
TXSCI_DAT	Sets transmit data	1 byte	100 bytes
RXSCI_DAT	Sets receive data	1 byte	100 bytes

Description of General Registers:

- Used as work register during data setting in all routines
- Used as RAM clear counter

- Sets RAM clear data

3.2.5 Flowcharts

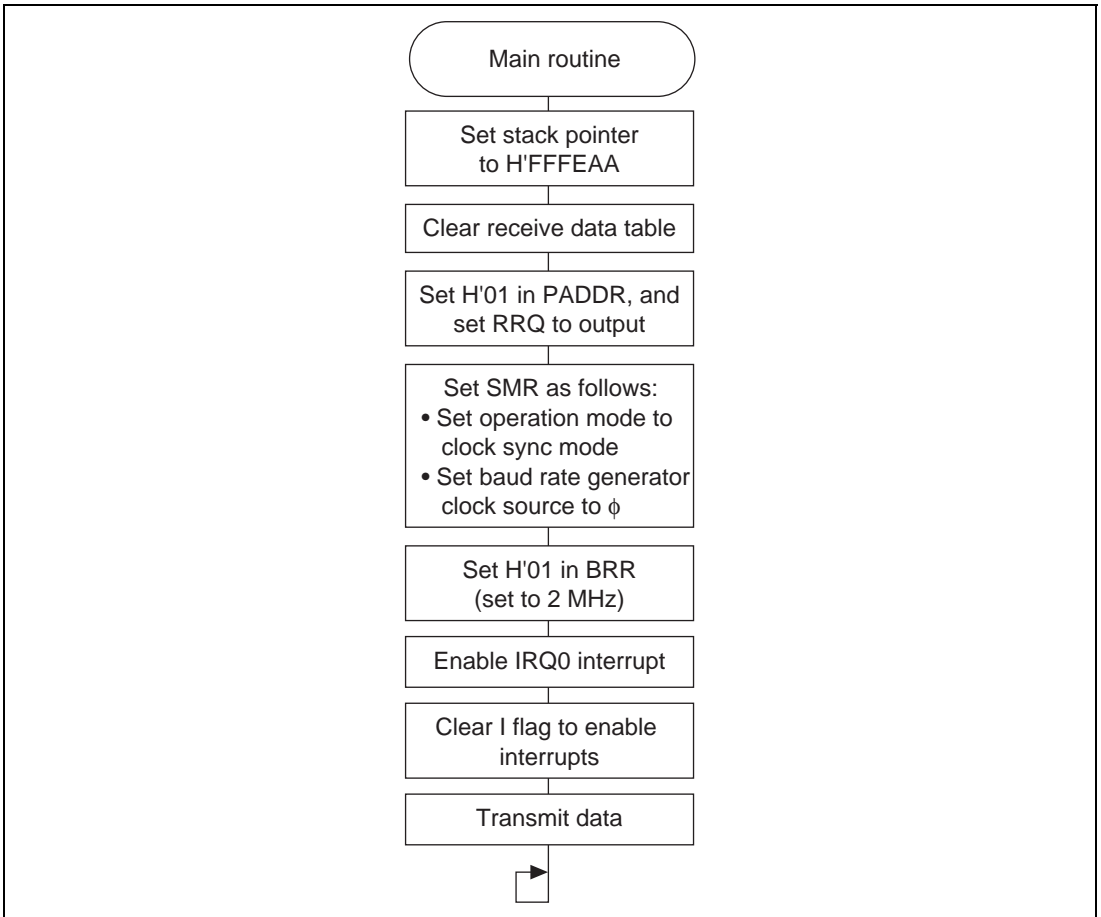


Figure 3.9 Main Routine Flowchart

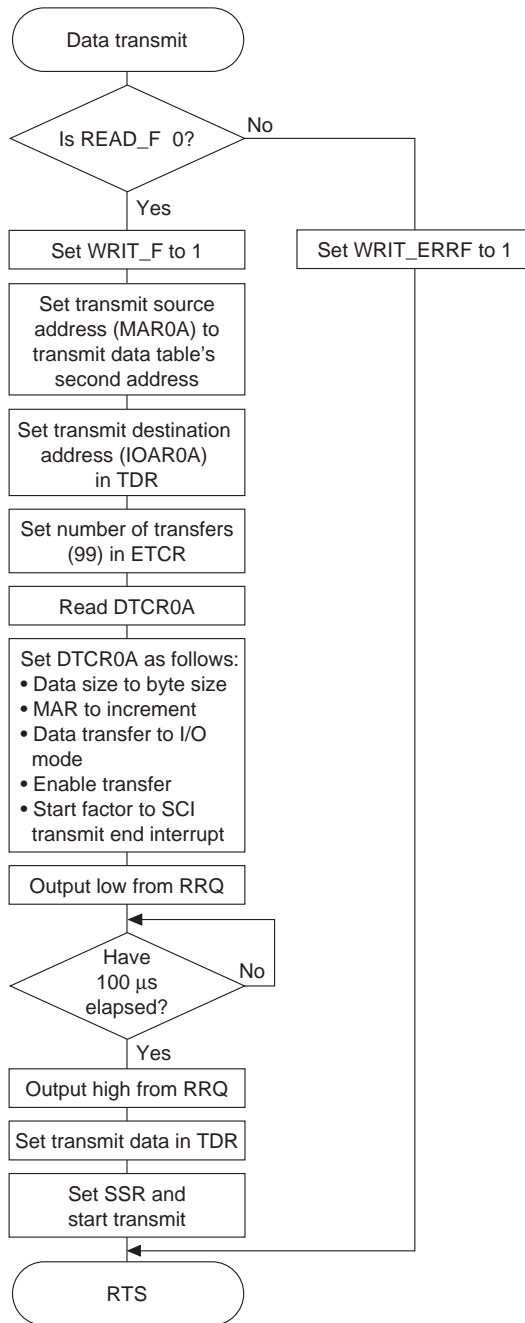


Figure 3.10 Data Transmit Flowchart

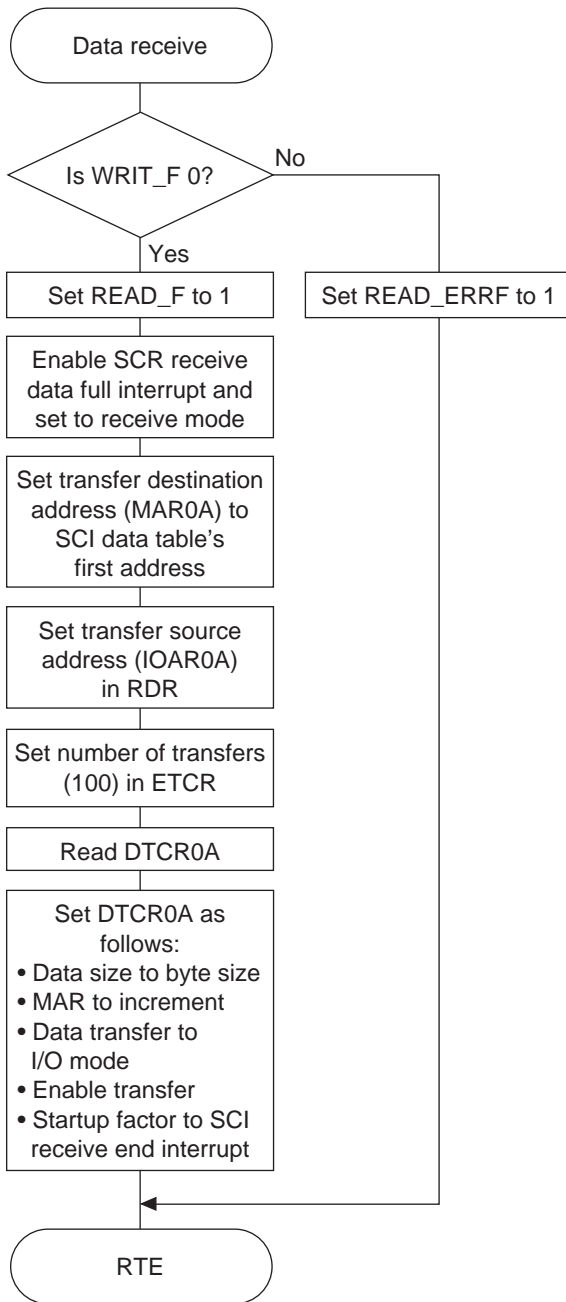


Figure 3.11 Data Receive Flowchart

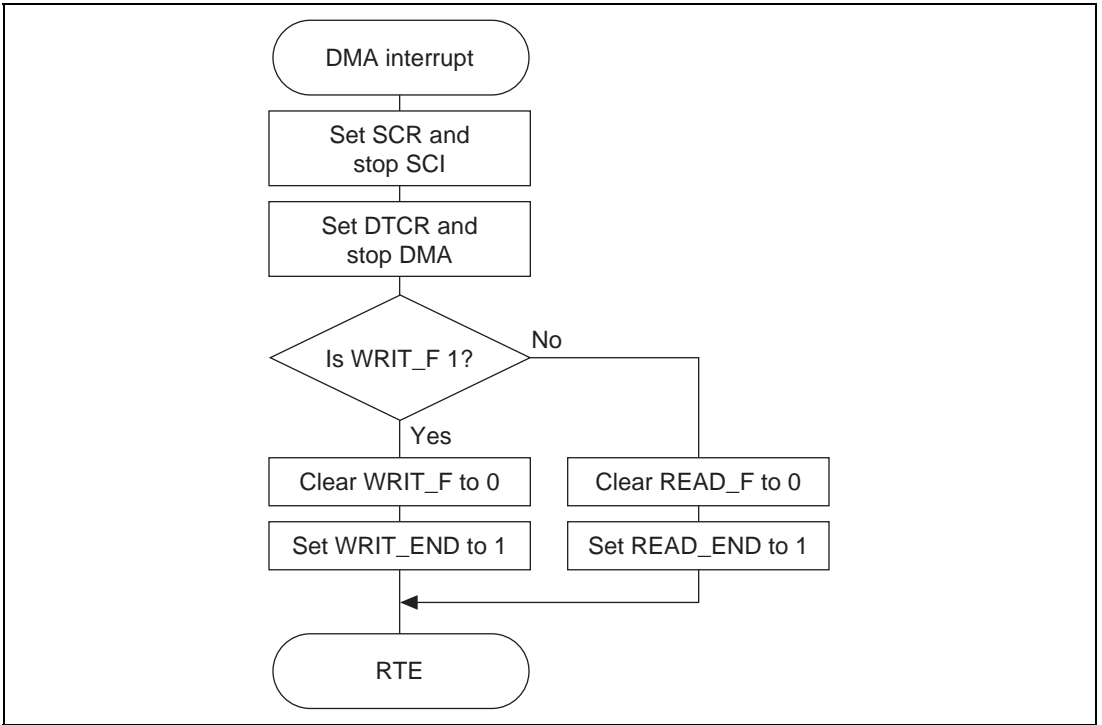


Figure 3.12 DMA Interrupt Flowchart

3.2.6 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7          7          .CPU      300HA
8 000000   8          .SECTION  VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES      .DATA.L   HSCIMN
11         11 ;
12 00001C   12          .ORG      H'00001C
13 00001C 00010000 13 NMI      .DATA.L   HSCIMN
14         14 ;
15 000030   15          .ORG      H'000030
16 000030 0001008E 16 IRQ0     .DATA.L   HSCIRX
17 000034 00010000 17 IRQ1     .DATA.L   HSCIMN
18 000038 00010000 18 IRQ2     .DATA.L   HSCIMN
19 00003C 00010000 19 IRQ3     .DATA.L   HSCIMN
20 000040 00010000 20 IRQ4     .DATA.L   HSCIMN
21 000044 00010000 21 IRQ5     .DATA.L   HSCIMN
22 000048 00010000 22 IRQ6     .DATA.L   HSCIMN
23 00004C 00010000 23 IRQ7     .DATA.L   HSCIMN
24 000050 00010000 24 WOVI0    .DATA.L   HSCIMN
25 000054 00010000 25 CMI      .DATA.L   HSCIMN
26         26 ;
27 000060   27          .ORG      H'000060
28 000060 00010000 28 IMIA0    .DATA.L   HSCIMN
29 000064 00010000 29 IMIB0    .DATA.L   HSCIMN
30 000068 00010000 30 OVI0     .DATA.L   HSCIMN
31         31 ;
32 000070   32          .ORG      H'000070
33 000070 00010000 33 IMIA1    .DATA.L   HSCIMN
34 000074 00010000 34 IMIB1    .DATA.L   HSCIMN
35 000078 00010000 35 OVI1     .DATA.L   HSCIMN
36         36 ;
37 000080   37          .ORG      H'000080
38 000080 00010000 38 IMIA2    .DATA.L   HSCIMN
39 000084 00010000 39 IMIB2    .DATA.L   HSCIMN
40 000088 00010000 40 OVI2     .DATA.L   HSCIMN
41         41 ;
42 000090   42          .ORG      H'000090
43 000090 00010000 43 IMIA3    .DATA.L   HSCIMN

```

```

44 000094 00010000      44 IMIB3      .DATA.L      HSCIMN
45 000098 00010000      45 OVI3       .DATA.L      HSCIMN
46                                     46 ;
47 0000A0                                     47          .ORG          H'0000A0
48 0000A0 00010000      48 IMIA4      .DATA.L      HSCIMN
49 0000A4 00010000      49 IMIB4      .DATA.L      HSCIMN
50 0000A8 00010000      50 OVI4       .DATA.L      HSCIMN
51                                     51 ;
52 0000B0                                     52          .ORG          H'0000B0
53 0000B0 000100CC      53 DEND0A     .DATA.L      DMAEND
54 0000B4 00010000      54 DEND0B     .DATA.L      HSCIMN
55 0000B8 00010000      55 DEND1A     .DATA.L      HSCIMN
56 0000BC 00010000      56 DEND1B     .DATA.L      HSCIMN
57 0000C0 00010000      57 DEND2A     .DATA.L      HSCIMN
58 0000C4 00010000      58 DEND2B     .DATA.L      HSCIMN
59 0000C8 00010000      59 DEND3A     .DATA.L      HSCIMN
60 0000CC 00010000      60 DEND3B     .DATA.L      HSCIMN
61 0000D0 00010000      61 ERI0       .DATA.L      HSCIMN
62 0000D4 00010000      62 RXI0       .DATA.L      HSCIMN
63 0000D8 00010000      63 TXI0       .DATA.L      HSCIMN
64 0000DC 00010000      64 TEI0       .DATA.L      HSCIMN
65 0000E0 00010000      65 ERI1       .DATA.L      HSCIMN
66 0000E4 00010000      66 RXI1       .DATA.L      HSCIMN
67 0000E8 00010000      67 TXI1       .DATA.L      HSCIMN
68 0000EC 00010000      68 TEI1       .DATA.L      HSCIMN
69 0000F0 00010000      69 ADI        .DATA.L      HSCIMN
70                                     70 ;
71                                     71 ;*****
72                                     72 ;*                                     *
73                                     73 ;*          RAM ALLOCATION          *
74                                     74 ;*                                     *
75                                     75 ;*****
76                                     76 ;
77 FFFFEA                                     77          .SECTION      RAM,DATA,LOCATE=H'FFFEA
78 FFFFEA 00000064      78 RXSCI_DAT  .RES.B      D'100
79                                     79 ;
80 FFFF0E 00000001      80 CSC_FL A   .RES.B      1          ;Flag area
81          00000000      81 WRIT_F     .EQU        0          ;Write action flag
82          00000001      82 READ_F     .EQU        1          ;Read action flag
83          00000002      83 WRIT_ERRF .EQU        2          ;Write error flag
84          00000003      84 READ_ERRF .EQU        3          ;Read error flag
85          00000004      85 WRIT_ENDF .EQU        4          ;Write end flag
86          00000005      86 READ_ENDF .EQU        5          ;Read end flag
87                                     87 ;

```

```

88 ;*****
89 ;*
90 ;*      SYMBOL DEFINITIONS
91 ;*
92 ;*****
93 ;
94 ;***** DMAC0A *****
95 00FFFF20 95 MAR0AR: .EQU      H'FFFF20 ;Memory address register
96 00FFFF24 96 ETCR0AH: .EQU      H'FFFF24 ;Transmit cunt register (H)
97 00FFFF25 97 ETCR0AL: .EQU      H'FFFF25 ;Transmit cunt register (L)
98 00FFFF26 98 IOAR0A: .EQU      H'FFFF26 ;I/O address register
99 00FFFF27 99 DTCR0A: .EQU      H'FFFF27 ;Data transfer control
;register
100 00000007 100 DTE: .EQU      7 ;Data transfer enable bit
101 ;
102 ;
103 ;***** SCI *****
104 00FFFFB0 104 SMR0: .EQU      H'FFFFB0 ;Serial mode register
105 00FFFFB1 105 BRR0: .EQU      H'FFFFB1 ;bit rate register
106 00FFFFB2 106 SCR0: .EQU      H'FFFFB2 ;Serial control register
107 00FFFFB3 107 TDR0: .EQU      H'FFFFB3 ;Transmit data register
108 00FFFFB4 108 SSR0: .EQU      H'FFFFB4 ;Serial status register
109 00000007 109 TDRE .EQU      7 ;Transmit data register
;enable bit
110 ;
111 ;***** I/O *****
112 00FFFFD1 112 PADDR: .EQU      H'FFFFD1 ;Port A data direction
;register
113 00FFFFD3 113 PADR: .EQU      H'FFFFD3 ;Port A data register
114 00000000 114 RRQ .EQU      0 ;RRQ signal bit
115 ;
116 ;***** INT CONTRAL *****
117 00FFFFF5 117 IER: .EQU      H'FFFFF5 ;IRQ enable register
118 00000000 118 IRQOE .EQU      0 ;IRQ0 enable bit
119 00FFFFF6 119 ISR: .EQU      H'FFFFF6 ;IRQ status register
120 00000000 120 IRQOF .EQU      0 ;IRQ0 status bit
121 ;
122 ;*****
123 ;*
124 ;*      MAIN PROGRAM : HSCIMN
125 ;*
126 ;*****
127 ;
128 010000 128 .SECTION  PROG, CODE, LOCATE=H'010000
129 ;
130 00010000 130 HSCIMN: .EQU      $

```

```

131 010000 7A0700FFFEAA 131          MOV.L      #H'FFFEAA,SP ;Initialize stack pointer
132                                     132 ;
133 010006 F100          133          MOV.B      #0,R1H          ;Counter clr
134 010008 F900          134          MOV.B      #0,R1L
135 01000A 7A0000FFFEAA 135          MOV.L      #RXSCI_DAT,ERO
136 010010 6889          136 HSCIMN01: MOV.B      R1L,@ERO
137 010012 0B70          137          INC.L      #1,ERO
138 010014 8101          138          ADD.B      #1,R1H
139 010016 A164          139          CMP.B      #D'100,R1H      ;Counter = 100 ?
140 010018 46F6          140          BNE       HSCIMN01      ;No
141                                     141 ;
142                                     142 ;***** Initialize I/O *****
143 01001A F801          143          MOV.B      #H'01,R0L
144 01001C 38D3          144          MOV.B      R0L,@PADR      ;Initialize PADR
145 01001E 38D1          145          MOV.B      R0L,@PADDR     ;Initialize PADDR(RRQ)
146                                     146 ;
147                                     147 ;***** Initialize SCIO *****
148 010020 F880          148          MOV.B      #H'80,R0L
149 010022 38B0          149          MOV.B      R0L,@SMR0      ;Initialize SMAR0
                                     ;(1 -> C/A)
150 010024 F801          150          MOV.B      #H'01,R0L
151 010026 38B1          151          MOV.B      R0L,@BRR0      ;SET BRR0(2M bit/s)
152                                     152 ;
153                                     153 ;***** Initialize INT CONTROLER *****
154 010028 7FF57000     154          BSET      #IRQ0E,@IER
155                                     155 ;
156 01002C 0700          156          LDC.B      #0.CCR          ;Enable interrupt
157                                     157 ;
158 01002E 5E010034     158          JSR       @HSCITX
159 010032 40FE          159 HSCIMN99: BRA       HSCIMN99
160                                     160 ;
161                                     161 ;*****
162                                     162 ;*
163                                     163 ;*      NAME : HSCITX(Output data)
164                                     164 ;*
165                                     165 ;*****
166                                     166 ;*
167                                     167 ;*      ENTRY : WRIT_F(Write action flag)
168                                     168 ;*      READ_ERRF(Read error flag)
169                                     169 ;*      RETURNS : READ_F(Read action flag)
170                                     170 ;*      WRIT_ERRF(Write error flag)
171                                     171 ;*
172                                     172 ;*****
173                                     173 ;
174 010034 7E0E7310     174 HSCITX:  BTST      #READ_F,@CSC_FLG;READ_F = 0 ?
175 010038 5860004C     175          BNE       HSCITX01      ;No

```

```

176 01003C 7F0E7000      176      BSET      #WRIT_F,@CSC_FL A;1 -> WRIT_F
177
178 010040 7A00000100FA      178      MOV.L     #TXSCI_DAT,ER0
179 010046 0B70              179      INC.L     #1,ER0
180 010048 01006B80FF20      180      MOV.L     ER0,@MAR0AR      ;Set MAR0A
181 01004E F8B3              181      MOV.B     #H'B3,R0L
182 010050 3826              182      MOV.B     R0L,@IOAR0A      ;Set IOAR0A(TDR0)
183 010052 79000063          183      MOV.W     #D'99,R0
184 010056 6B80FF24          184      MOV.W     R0,@ETCR0AH      ;Store tranfer count
185 01005A 2827              185      MOV.B     @DTCR0A,R0L
186 01005C F88C              186      MOV.B     #H'8C,R0L
187 01005E 3827              187      MOV.B     R0L,@DTCR0A      ;Initialize DTCR0A
                                ;(1 -> DTIE, DTS2A)

188
189 010060 7FD37200          189      BCLR     #RRQ,@PADR      ;RRQ -> "L" output
190 010064 F100              190      MOV.B     #0,R1H          ;Counter clr
191 010066 8101          191 HSCITX02: ADD.B     #H'01,R1H
192 010068 A1C8              192      CMP.B     #H'C8,R1H      ;100us pass?
193 01006A 46FA          193      BNE      HSCITX02
194
195 01006C 7FD37000          195      BSET     #RRQ,@PADR      ;RRQ -> "H" output
196
197 010070 7EB47370          197 HSCITX03: BTST     #TDRE,@SSR0      ;TDRE = 1?
198 010074 47FA          198      BEQ      HSCITX03      ;No
199 010076 6A28000100F4      199      MOV.B     @TXSCI_DAT,R0L
200 01007C 38B3          200      MOV.B     R0L,@TDR0      ;Set TDR
201 01007E 7FB47270          201      BCLR.B   #7,@SSR0
202 010082 F8A0          202      MOV.B     #H'A0,R0L
203 010084 38B2          203      MOV.B     R0L,@SCR0      ;Initialize SCR0
                                ;(1 -> TIE,TE)

204 010086 5470          204 HSCITX99: RTS
205
206 010088 7F0E7020          206 HSCITX01: BSET     #WRIT_ERRF,@CSC_FL A;1 -> WRIT_ERRF
207 01008C 40F8          207      BRA      HSCITX99
208
209
210
211
212
213
214
215
216
217
218
219

```



```

220 ;*****
221 ;
222 01008E 01006DF0 222 HSCIRX: PUSH.L ERO ;Escape register
223 010092 7E0E7300 223 BTST #WRIT_F,@CSC_FL A;WRIT_F = 0?
224 010096 5860002C 224 BNE HSCIRX01 ;No
225 01009A 7F0E7010 225 BSET #READ_F,@CSC_FL A;1 -> READ_F
226 01009E F852 226 MOV.B #H'52,R0L
227 0100A0 38B2 227 MOV.B R0L,@SCR0 ;Initialize SCR0
; (1 -> RIE,RE,CKE1)

228 ;
229 0100A2 7A0000FFFEAA 229 MOV.L #RXSCI_DAT,ERO
230 0100A8 01006B80FF20 230 MOV.L ERO,@MAR0AR ;Set MAR0A
231 0100AE F8B5 231 MOV.B #H'B5,R0L
232 0100B0 3826 232 MOV.B R0L,@IOAR0A ;Set IOAR0A(RDR0)
233 0100B2 F864 233 MOV.B #D'100,R0L
234 0100B4 3825 234 MOV.B R0L,@ETCR0AL ;Store tranfer count
235 0100B6 2827 235 MOV.B @DTCR0A,R0L ;Read DTCR0A
236 0100B8 F80D 236 MOV.B #H'0D,R0L
237 0100BA 3827 237 MOV.B R0L,@DTCR0A ;Initialize DTCR0A
; (1->DTIE,DTS2A,DTS0A)

238 0100BC 7F277070 238 BSET #DTE,@DTCR0A ;Start DMAC0A
239 ;
240 0100C0 01006D70 240 HSCIRX99: POP.L ERO
241 0100C4 5670 241 RTE
242 ;
243 0100C6 7F0E7030 243 HSCIRX01: BSET #READ_ERRF,@CSC_FL A;1 -> READ_ERRF
244 0100CA 40F4 244 BRA HSCIRX99
245 ;
246 ;*****
247 ;* *
248 ;* NAME : DMAEND(Set next data) *
249 ;* *
250 ;*****
251 ;* *
252 ;* ENTRY : WRIT_F(Write action flag) *
253 ;* RETURNS : WRIT_F(Write action flag) *
254 ;* READ_F(Read action flag) *
255 ;* WRIT_ENDF(Write end flag) *
256 ;* READ_ENDF(Read end flag) *
257 ;* *
258 ;*****
259 ;
260 0100CC F800 260 DMAEND: MOV.B #'00,R0L
261 0100CE 38B2 261 MOV.B R0L,@SCR0
262 0100D0 3827 262 MOV.B R0L,@DTCR0A ;Initialize DTCR0A
; (0 -> DTIE,DTS2A)

```

```

263 0100D2 7E0E7300      263      BTST      #WRIT_F,@CSC_FL A;WRIT_F = 1?
264 0100D6 5870000C      264      BEQ      DMAEND01      ;No
265      265 ;
266 0100DA 7F0E7200      266      BCLR      #WRIT_F,@CSC_FL A;WRIT_F Clear
267 0100DE 7F0E7040      267      BSET      #WRIT_ENDF,@CSC_FL A ;1 -> WRIT_ENDF
268 0100E2 5800000C      268      BRA      DMAEND99
269      269 ;
270 0100E6 7F0E7210      270 DMAEND01: BCLR      #READ_F,@CSC_FL A;READ_F Clear
271 0100EA 7F0E7050      271      BSET      #READ_ENDF,@CSC_FL A ;1 -> READ_ENDF
272 0100EE 7FF67200      272      BCLR      #IRQ0F,@ISR      ;IRQ0F Clr
273      273 ;
274 0100F2 5670      274 DMAEND99 RTE
275      275 ;
276      276 ;
277      277 ;*****
278      278 ;* *
279      279 ;*      DATA TABLE (WRITE DATA TABLE) *
280      280 ;* *
281      281 ;*****
282 0100F4 010203040506070B 282 TXSCI_DAT .DATA.B      D'01,D'02,D'03,D'04,D'05,D'06,
      D'07,D'08,D'09,D'10,
283 0100FC 090A0B0C0D0E0F10 283 *      D'11,D'12,D'13,D'14,D'15,D'16,
      D'17,D'18,D'19,D'20,
284 010104 1112131415161718 284 *      D'21,D'22,D'23,D'24,D'25, D'26,
      D'27,D'28,D'29,D'30,
285 01010C 191A1B1C1D1E1F20 285 *      D'31,D'32,D'33,D'34,D'35,D'36,
      D'37,D'38,D'39,D'40
286 010114 2122232425262728 286 *      D'41,D'42,D'43,D'44,D'45,D'46,
      D'47,D'48,D'49,D'50,
287 01011C 292A2B2C2D2E2F30 287 *      D'51,D'52,D'53,D'54,D'55,D'56,
      D'57,D'58,D'59,D'60,
288 010124 3132333435363738 288 *      D'61,D'62,D'63,D'64,D'65,D'66,
      D'67,D'68,D'69,D'70,
289 01012C 393A3B3C3D3E3F40 289 *      D'71,D'72,D'73,D'74,D'75,D'76,
      D'77,D'78,D'79,D'80,
290 010134 4142434445464748 290 *      D'81,D'82,D'83,D'84,D'85,D'86,
      D'87,D'88,D'89,D'90
291 01013C 494A4B4C4D4E4F50 291 *      D'91,D'92,D'93,D'94,D'95,D'96,
      D'97,D'98,D'99,D'100
      010144 5152535455565758
      01014C 595A5B5C5D5E5F60
      010154 61626364
292      292
293      293      .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0

```

3.3 Sine Wave Output

MCU: H8/3042

Function: ITU, DMAC, D/A

3.3.1 Specifications

- Output of a waveform corresponding with a 100 bit binary digit string (1010111...) set in RAM (see figure 3.13)
- 1-bit data set in RAM outputs a sine waveform with the following frequencies:
 - 0: 2400 Hz full sine wave output
 - 1: 1200 Hz half sine wave output
- Data set in RAM is output MSB first
- Applying this sample task allows MSK modem functions

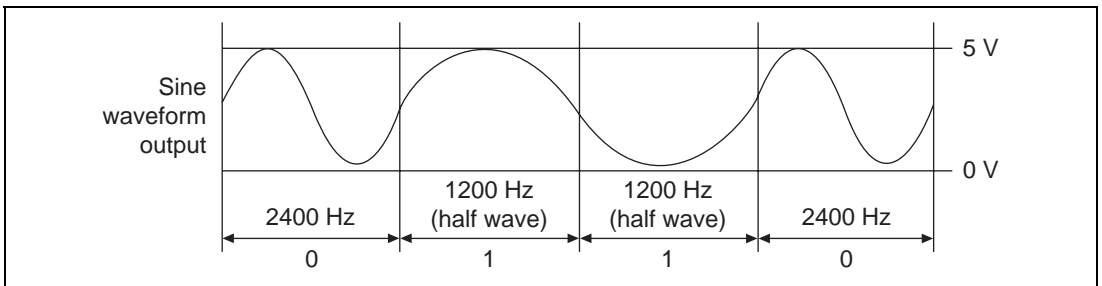


Figure 3.13 Example of Sine Waveform Output

3.3.2 Method

Sine Wave Output Operation: Figure 3.14 shows an example of sine waveform output employing D/A converter output. The following is an overview of the operation.

- Analog voltages are output using the D/A converter (see figure 3.10)
- D/A converter data for one cycle of the sine wave corresponding with the output analog voltages is stored in a ROM data table
- D/A data stored in the data table is transferred to the D/A converter for each output cycle (T), and is converted to a step-form voltage wave
- This stepped waveform is passed through filter circuitry to give the sine wave output indicated by the dotted line in figure 3.14

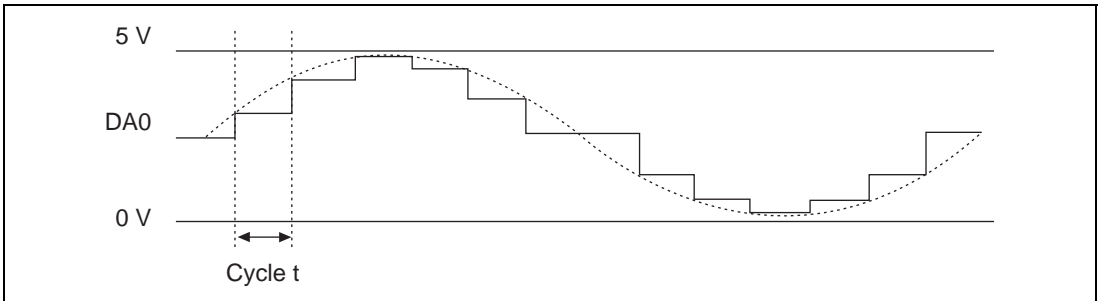


Figure 3.14 Example of Sine Waveform Output

Data Format: Figure 3.15 shows an example of waveform output matched with a 0101 4-bit bit pattern.

As shown in the figure, a 1 output becomes a 1200 Hz half-wave output. Consequently, the next output waveform must be either a 2400 Hz waveform or a negative phase 1200 Hz waveform.

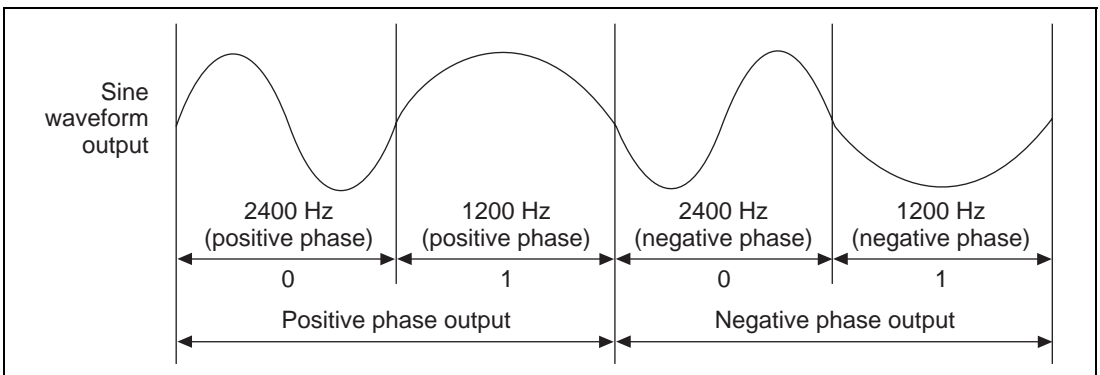


Figure 3.15 "0101" 4-Bit Data Output Example

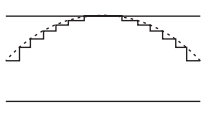

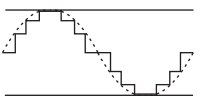
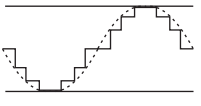
1200 Hz and 2400 Hz Positive/Negative Waveform Output Method: Figure 3.16 shows a waveform data table used for each waveform output. The table stores 1200 Hz positive half-wave, 1200 Hz negative half-wave and 2400 Hz full-wave data.

Table 3.12 gives output waveforms and methods of using the waveform data table. As shown in the table, waveforms are output by judging the waveform data table's start address and address increment and decrement.

Sine waveform data table ROM	Address
1200-Hz half-wave data (positive phase)	S12USTADR (1200-Hz positive phase half-wave data table first address)
1200-Hz half-wave data (negative phase)	S12LSTADR (1200-Hz negative phase half-wave data table first address)
2400-Hz full-wave data	S24STADR (2400-Hz data table first address) S24EDADR (2400-Hz data table last address)

Figure 3.16 Correlation of Sine Waveform Data Table and Output Waveform

Table 3.12 Output Waveforms and Methods of Using Waveform Data Tables

Waveform Designation	Output Waveform	Start Address	Increment/Decrement	Content
1 output		S12USTADR	Increment	Increment from data table S12USTADR to S12LSTADR-1 address and output of data
1 output		S12LSTADR	Increment	Increment from data table S12LSTADR to S24STADR-1 address and output of data
0 output		S24STADR	Increment	Increment from data table S24STADR to S24EDADR and output of data
0 output		S24EDADR	Decrement	Decrement from data table S24EDADR to S24STADR and output of data

3.3.3 Description of Functions Used

Figure 3.17 shows the H8/3042 on-chip functions used in this sample task. The following H8/3042 functions are used for sine wave output.

Sine Wave Data Table: Sets sine wave data table in ROM for D/A conversion.

ITU: Clears timer counter each time compare/match A occurs. Also starts up DMAC0A.

DMAC0A: Starts up by ITU compare/match A, and transfers output data from sine wave data table to D/A converter's DADR.

D/A: Converts sine wave data table digital input to analog output. Can also set analog voltage range using AVCC as reference voltage.

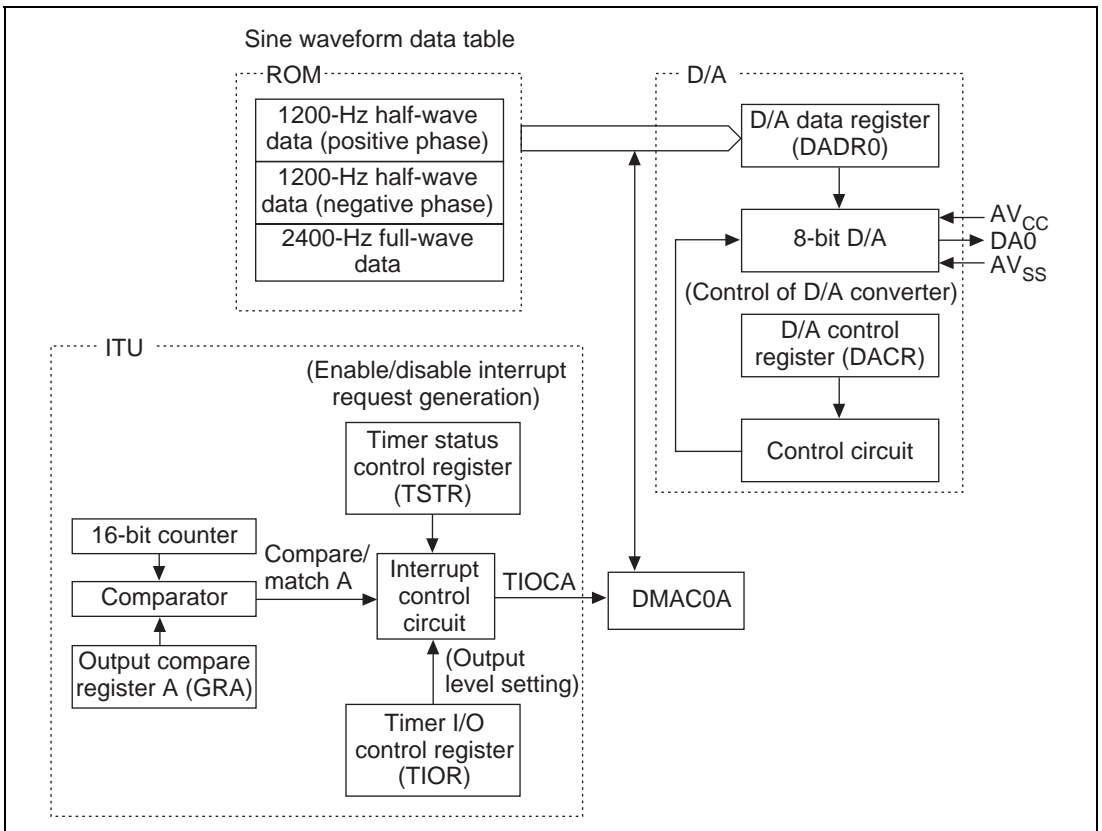


Figure 3.17 Block Diagram of Sine Wave Output Circuit

Table 3.13 shows the H8/3042 function allocation for sine wave output with this sample task.

Table 3.13 H8/3042 Function Allocation

H8/3042 Function	Function	
ITU (ch0)	TCNT0	16-bit counter generating compare/match with constant cycle
	GRA0	Output compare register
	TCR0	Selects counter clock and counter clear factor
	TIOR0	Sets GRA to output compare register
	TSR0	Indicates compare/match and overflow status
	TIER0	Selects enable/disable interrupts
	TSTR	Enables/disables timer counter operation
DMAC0A	DTCR0A	Controls DMAC0A operation
	MAR0A	Sets start address of data table corresponding with sine wave
	IOAR0A	Sets DADR address
	ETCR0A	Sets number of transfers
D/A	DADR	Stores data for conversion
	DACR	Controls D/A converter operation
	AV _{cc}	Analog block power supply and reference voltage
	AV _{ss}	Analog block ground and reference voltage
	DA0	Analog output

3.3.4 Description of Operations

Figure 3.18 shows the principle of operation. Sine waveforms are output due to H8/3042 hardware and software processing.

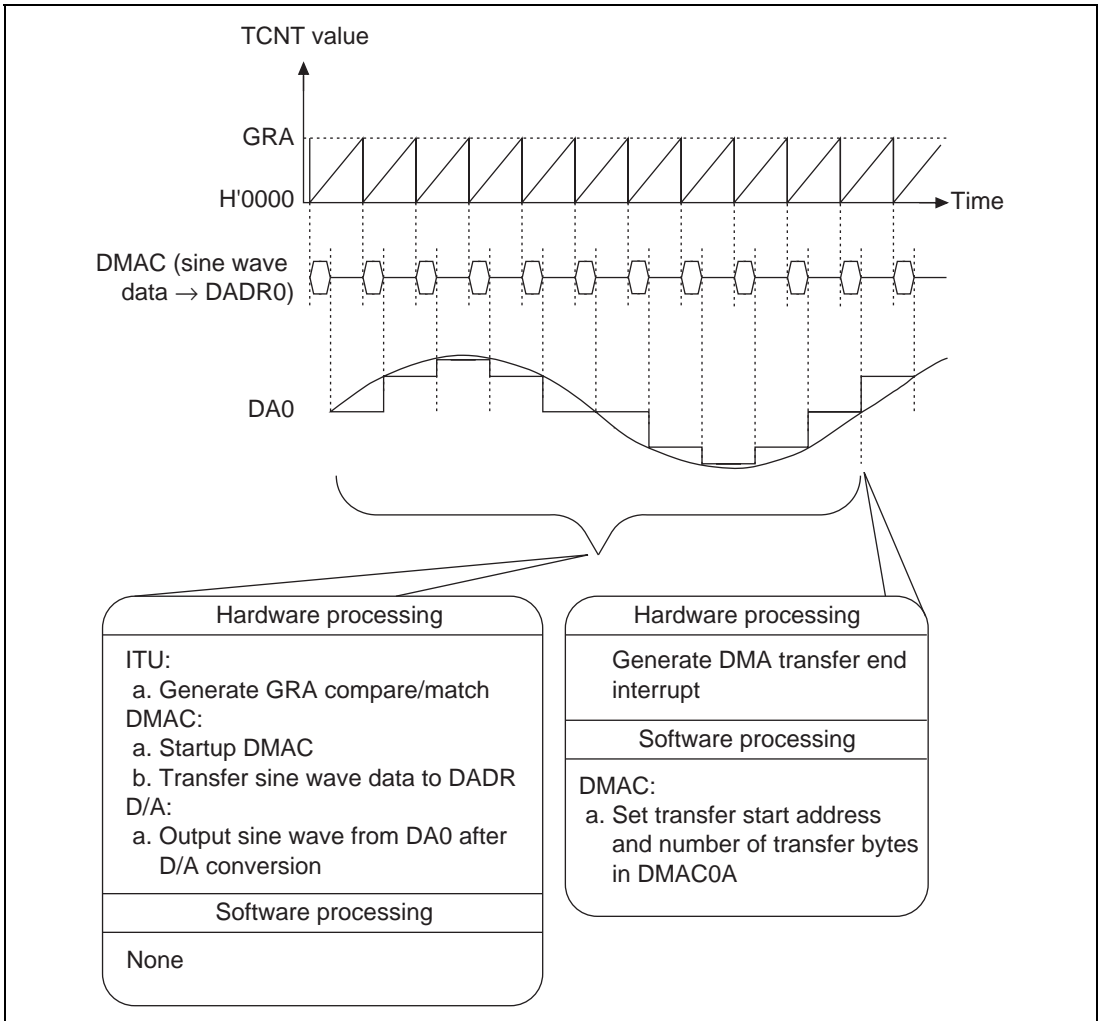


Figure 3.18 Principle of Sine Wave Output Operation (1)

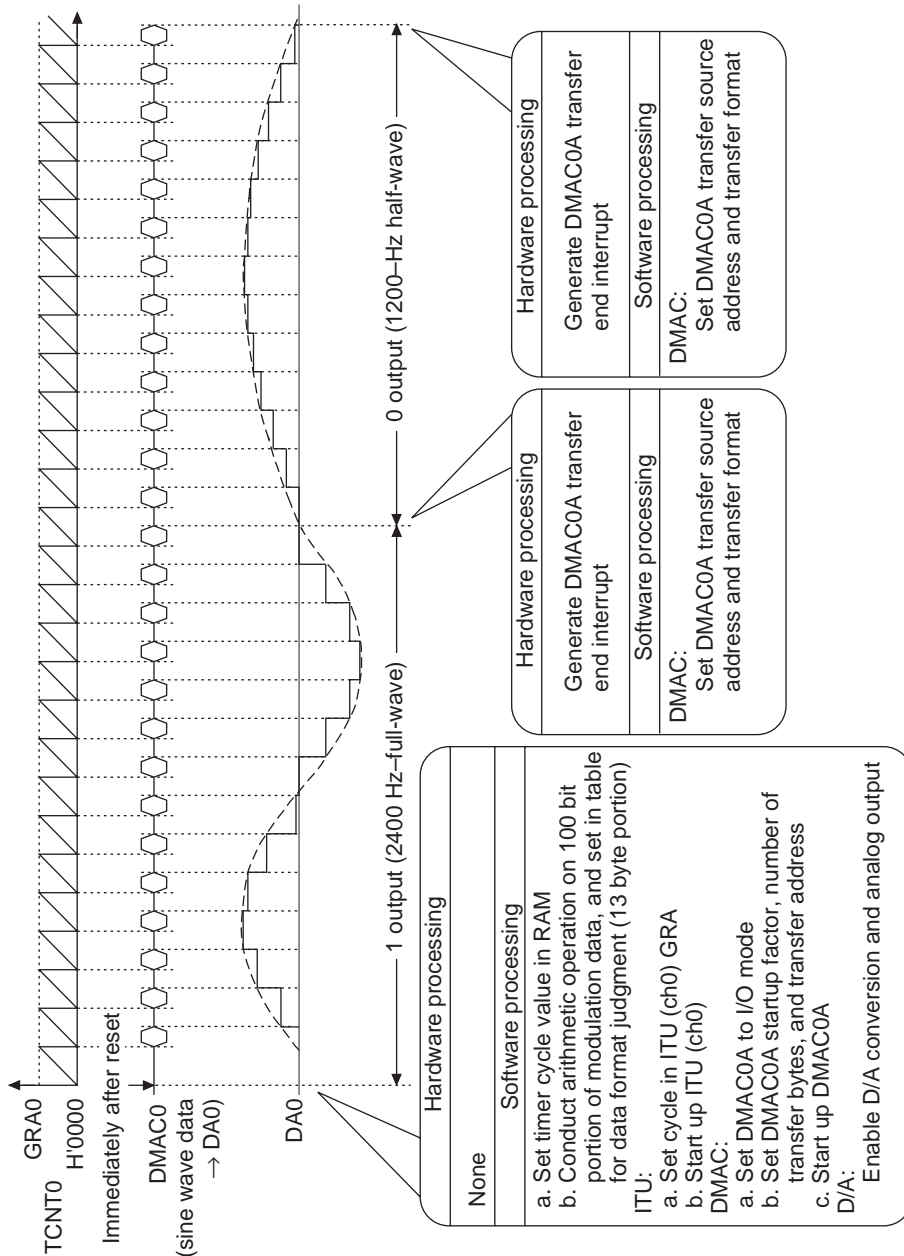


Figure 3.18 Principle of Sine Wave Output Operation (2)

3.3.5 Description of Software

The software for sine wave output is described below.

Table 3.14 Description of Modules

Module Name	Label Name	Function
Main routine	SINMN	Initial setting of stack pointer, 16-bit timer (ITU), DMAC and D/A converter
DMAC start	SINWAVEA	Startup by DMAC0A transfer end interrupt, and performs DMAC0A resetting and reactivation

Table 3.15 Description of Internal Registers

On-Chip Function	Register Name	Function	
ITU	GRA0	Sets data output cycle	
	TIER0	Enables IMFA interrupts	
	TCR0	Sets ITU as follows: <ul style="list-style-type: none"> • Clear counter with GRA compare/match • Count up with internal clock ϕ 	
	TIOR0	Sets GRA to output compare register and prohibits pin output	
	TSTR0	Enables TCNT counting operation	
	DMAC0A	DTCR0A	Sets DMAC0A as follows: <ul style="list-style-type: none"> • Data size to byte size • MAR to increment or decrement (differs depending on the module) • Data transfer to I/O mode • Transfer enable or disable (differs depending on the module) • Startup factor to ITU ch0 compare/match A
MAR0A		Sets sine waveform data table's transfer source address	
IOAR0A		Sets DADR address (transfer destination)	
ETCR0A		Sets number of transfers	
D/A		DADR	Stores data for D/A conversion (sine waveform data table)
		DACR	Sets D/A converter as follows: <ul style="list-style-type: none"> • Enable D/A converter channel 0 • Enable analog output DA0

Table 3.16 Description of Arguments

Label Name, Register Name	Function	Data Length	Module Name	I/O
SIN_MOD	Flag indicating waveform type currently being output.	1 byte	Main routine	Output
	H'01: "1" output (positive phase)		SINWAVE	I/O
	H'02: "1" output (negative phase)			
	H'04: "0" output (positive phase)			
	H'08: "0" output (negative phase)			
SIN_LWCOUN	Up counter counting up each time a DMA interrupt occurs. Processing of sine waveform data after counting 32 times.	1 byte	Main routine	I/O
			SINWAVE	I/O
SIN_FCOUNT	Up counter counting up each time a DMA interrupt occurs. Peripheral functions stopped after counting 100 times.	1 byte	Main routine	Input
			SINWAVE	I/O
SIN_ADR	Indicates address of next modulation data to be used. SINADR increments when SINLWDAT reads modulation data, and indicates next modulation data address.	1 long word	Main routine	Input
			SINWAVE	I/O
SIN_DAT	Stores current modulation data.	1 long word	Main routine	I/O
	1 bit shift and fetch modulation data.		SINWAVE	I/O
	After fetching all 32 bits of modulation data, modulation data is read in from the address indicated by SINADR.			

Table 3.17 Description of RAM

Label Name	Function	Data Length	Module Name
OUT_DAT	Sets modulation data (1010111...) in RAM	100 bit	SINWAVEA

Table 3.18 Description of Data Tables

Table Name	Function	Data Length	Data Volume
S12USTADR	Sets sine waveform data for D/A conversion	Byte	60 bytes
S12LSTADR			
S24STADR			
S24EDADR			

Description of General Registers: Figure 3.19 describes the general registers.

	31	ERn	RnH	RnL	0
0	Used as work register during data setting in all routines				
1			Used as work register during interrupt processing		
2					
3					
4					
5					
6					
7					

Figure 3.19 Description of General Registers

3.3.6 Flowcharts

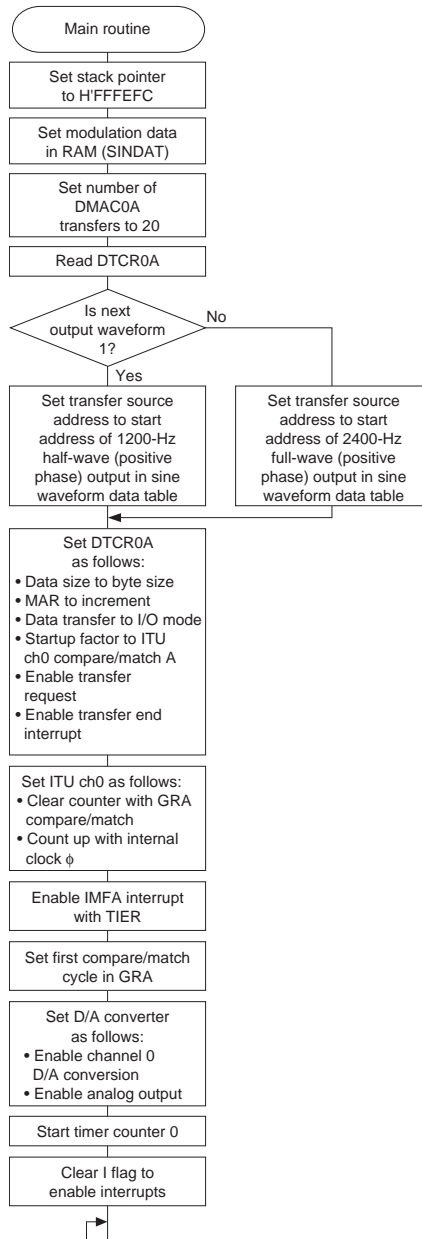


Figure 3.20 Main Routine Flowchart

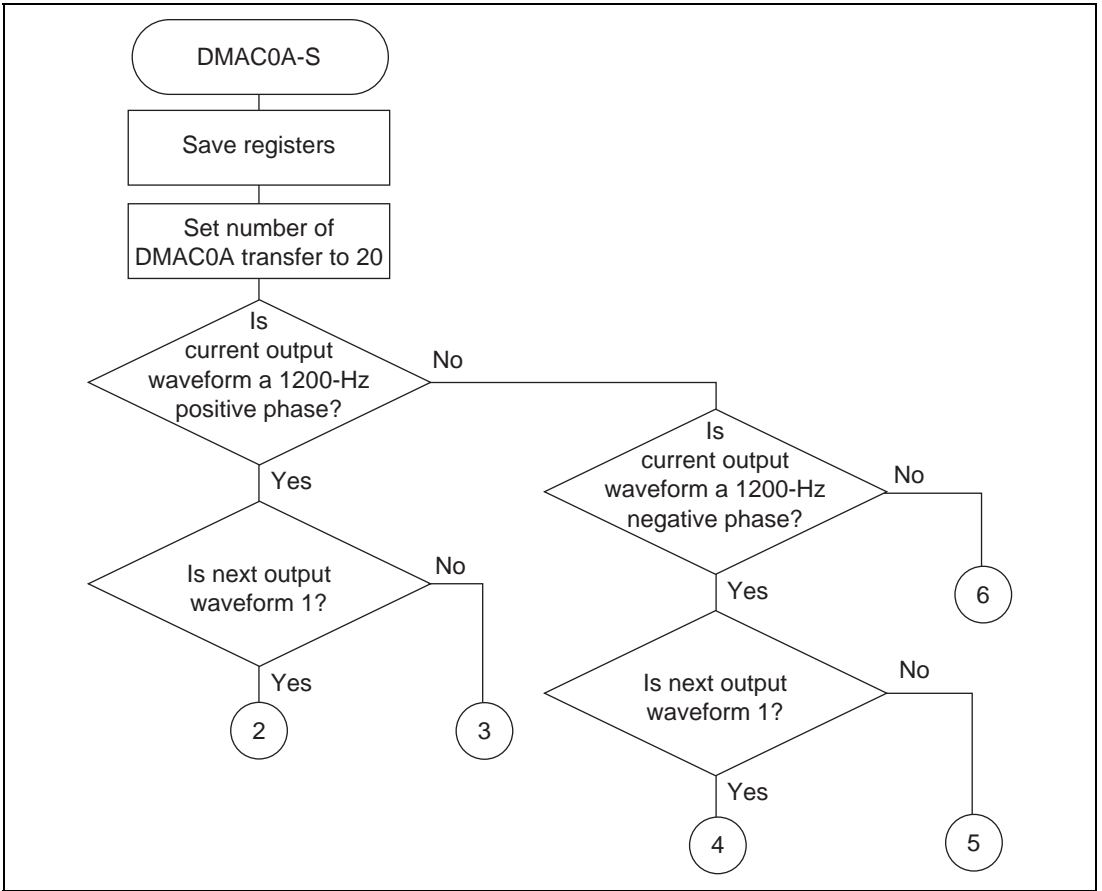


Figure 3.21 DMAC Start Flowchart (1)

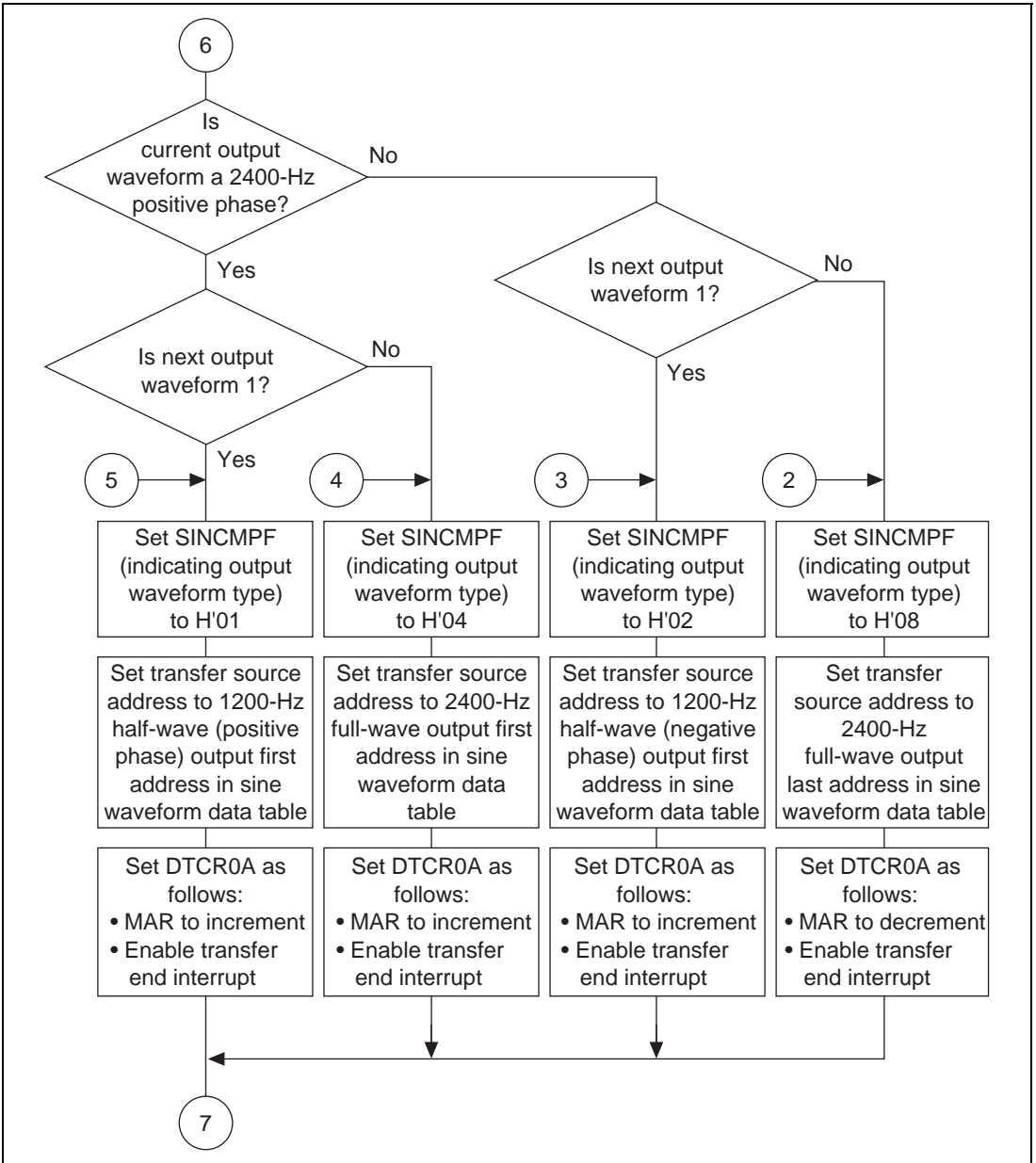


Figure 3.21 DMAC Start Flowchart (2)

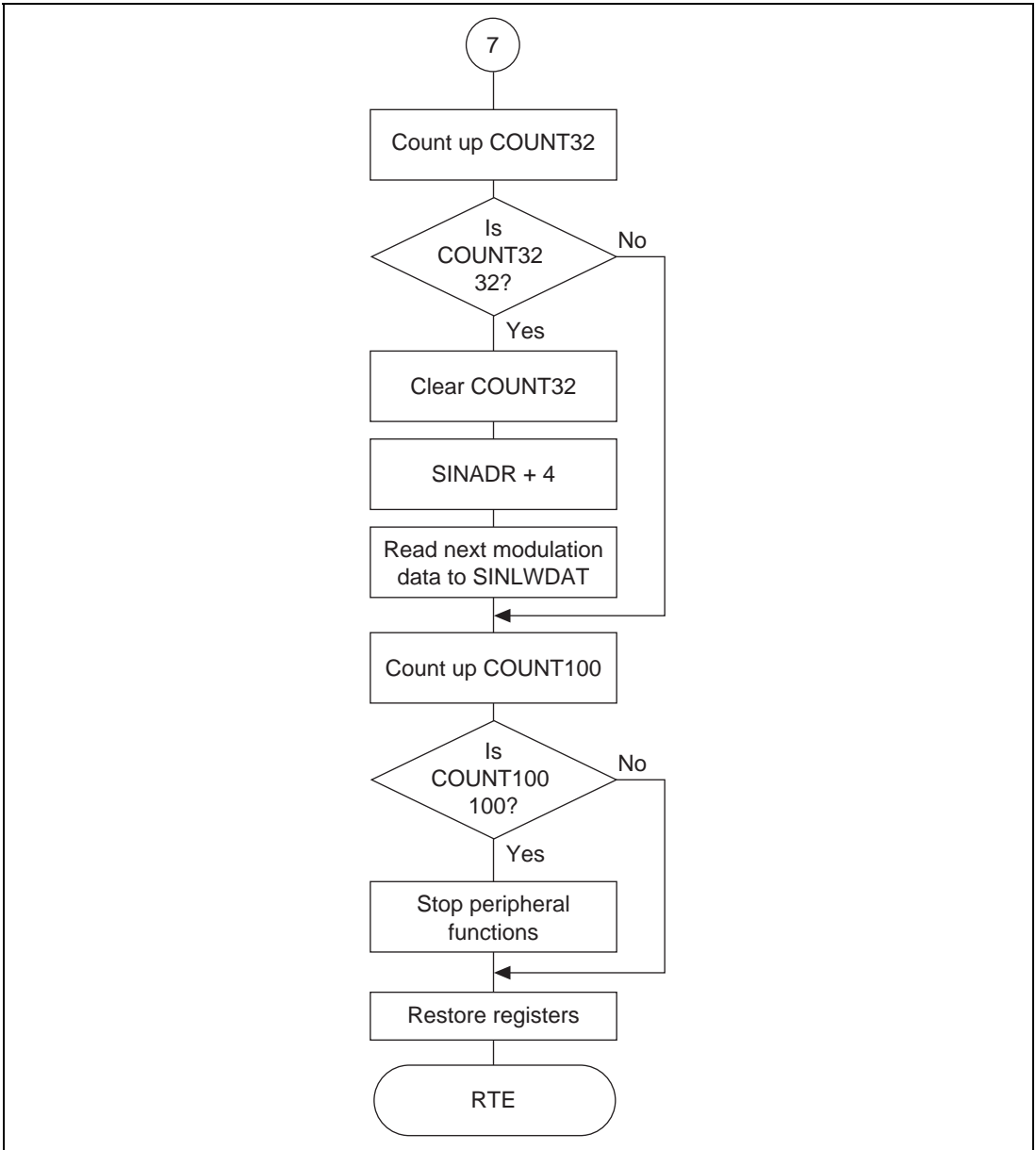


Figure 3.21 DMAC Start Flowchart (3)

3.3.7 Program list

```
1 ;*****
2 ;*
3 ;* VECTOR ADDRESS
4 ;*
5 ;*****
6 ;
7 .CPU 300HA
8 000000 .SECTION VECT, CODE, LOCATE=H'000000
9 ;
10 000000 00010000 10 RES .DATA.L SINMN
11 ;
12 00001C 12 .ORG H'00001C
13 00001C 00010000 13 NMI .DATA.L SINMN
14 ;
15 000030 15 .ORG H'000030
16 000030 00010000 16 IRQ0 .DATA.L SINMN
17 000034 00010000 17 IRQ1 .DATA.L SINMN
18 000038 00010000 18 IRQ2 .DATA.L SINMN
19 00003C 00010000 19 IRQ3 .DATA.L SINMN
20 000040 00010000 20 IRQ4 .DATA.L SINMN
21 000044 00010000 21 IRQ5 .DATA.L SINMN
22 ;
23 000050 23 .ORG H'000050
24 000050 00010000 24 WOVI .DATA.L SINMN
25 000054 00010000 25 CMI .DATA.L SINMN
26 ;
27 000060 27 .ORG H'000060
28 000060 00010000 28 TGAI0 .DATA.L SINMN
29 000064 00010000 29 TGBB0 .DATA.L SINMN
30 000068 00010000 30 TOVI0 .DATA.L SINMN
31 ;
32 000070 32 .ORG H'000070
33 000070 00010000 33 TGAI1 .DATA.L SINMN
34 000074 00010000 34 TGBI1 .DATA.L SINMN
35 000078 00010000 35 TOVI1 .DATA.L SINMN
36 ;
37 000080 37 .ORG H'000080
38 000080 00010000 38 TGAI2 .DATA.L SINMN
39 000084 00010000 39 TGBI2 .DATA.L SINMN
40 000088 00010000 40 TOVI2 .DATA.L SINMN
41 ;
42 000090 42 .ORG H'000090
43 000090 00010000 43 TGAI3 .DATA.L SINMN
```

```

44 000094 00010000      44  TGBI3      .DATA.L      SINMN
45 000098 00010000      45  TOVI3      .DATA.L      SINMN
46                                     46  ;
47 0000A0                                     47          .ORG          H'0000A0
48 0000A0 00010000      48  TGAI4      .DATA.L      SINMN
49 0000A4 00010000      49  TGBI4      .DATA.L      SINMN
50 0000A8 00010000      50  TOVI4      .DATA.L      SINMN
51                                     51  ;
52 0000B0                                     52          .ORG          H'0000B0
53 0000B0 000100AC      53  DEND0      .DATA.L      SINWAVE
54 0000B4 00010000      54  DEND1      .DATA.L      SINMN
55 0000B8 00010000      55  DEND2      .DATA.L      SINMN
56 0000BC 00010000      56  DEND3      .DATA.L      SINMN
57                                     57  ;
58 0000D0                                     58          .ORG          H'0000D0
59 0000D0 00010000      59  ERI0       .DATA.L      SINMN
60 0000D4 00010000      60  RXI0       .DATA.L      SINMN
61 0000D8 00010000      61  TXI0       .DATA.L      SINMN
62 0000DC 00010000      62  TEND0      .DATA.L      SINMN
63 0000E0 00010000      63  ERI1       .DATA.L      SINMN
64 0000E4 00010000      64  RXI1       .DATA.L      SINMN
65 0000E8 00010000      65  TXI1       .DATA.L      SINMN
66 0000EC 00010000      66  TEND1      .DATA.L      SINMN
67 0000F0 00010000      67  ADI        .DATA.L      SINMN
68                                     68  ;
69                                     69  ;*****
70                                     70  ;*                                     *
71                                     71  ;*          RAM ALLOCATION          *
72                                     72  ;*                                     *
73                                     73  ;*****
74                                     74  ;
75 FFFEFC                                     75          .SECTION      RAM,DATA,LOCATE=H'FFFEFC
76                                     76  ;
77 FFFEFC 00000004      77  SIN_ADR    .RES.L      1          ;Address of outupt data
78 FFFF00 00000004      78  SIN_DAT    .RES.L      1          ;Next data buff
79 FFFF04 00000001      79  SIN_MOD    .RES.B      1          ;Sin wave type
80 FFFF05 00000001      80  SIN_LWCOUN .RES.B      1          ;Count of 1 long ward data
81 FFFF06 00000001      81  SIN_FCOUNT .RES.B      1          ;Count of 100 bits data
82                                     82  ;
83 FFFF07 0000000D      83  OUT_DAT    .RES.B      D'13      ;Output data buff
84                                     84  ;
85                                     85  ;*****
86                                     86  ;*                                     *
87                                     87  ;*          SYMBOL DEFINITIONS          *
88                                     88  ;*                                     *
89                                     89  ;*****

```

```

90          90 ;
91          91 ;***** DMAC0A *****
92          00FFFF20 92 MAR0AR: .EQU      H'FFFF20 ;Memory address register
93          00FFFF24 93 ETCR0AH: .EQU      H'FFFF24 ;Transmit cunt register
94          00FFFF26 94 IOAR0A: .EQU      H'FFFF26 ;I/O address register
95          00FFFF27 95 DTCR0A: .EQU      H'FFFF27 ;Data transfer control
                                     ;register
96          00000007 96 DTE:      .EQU      7          ;Data transfer enable bit
97          97 ;
98          98 ;***** ITU *****
99          00FFFF60 99 TSTR:      .EQU      H'FFFF60 ;Timer start register
100         100 ;
101         101 ;***** ITU ch0 *****
102         00FFFF64 102 TCR0:      .EQU      H'FFFF64 ;timer control register
103         00FFFF65 103 TIOR0:     .EQU      H'FFFF65 ;timer I/O control register
104         00FFFF66 104 TIER0:     .EQU      H'FFFF66 ;timer interrupt enable
                                     ;register
105         00FFFF67 105 TSR0:      .EQU      H'FFFF67 ;timer status register
106         00FFFF68 106 TCNT0:     .EQU      H'FFFF68 ;timer counter register
107         00FFFF6A 107 GRA0H:     .EQU      H'FFFF6A ;General register
108         108 ;
109         109 ;***** D/A *****
110         00FFFFDE 110 DACR:      .EQU      H'FFFFDE ;D/A control register
111         111 ;
112         112 ;*****
113         113 ;*
114         114 ;*      MAIN PROGRAM : SINMN
115         115 ;*
116         116 ;*****
117         117 ;*
118 010000 118          .SECTION  PROG,CODE,LOCATE=H'010000
119         119 ;
120         00010000 120 SINMN:     .EQU      $
121 010000 7A0700FFFEFC 121          MOV.L     #H'FFFEFC,SP ;Initialize stack pointer
122         122 ;
123 010006 F8F0 123          MOV.B     #H'F0,R0L ;Initialize RAM data
124 010008 3807 124          MOV.B     R0L,@OUT_DAT
125 01000A F8F0 125          MOV.B     #H'F0,R0L
126 01000C 3808 126          MOV.B     R0L,@OUT_DAT+1
127 01000E F8F0 127          MOV.B     #H'F0,R0L
128 010010 3809 128          MOV.B     R0L,@OUT_DAT+2
129 010012 F8F0 129          MOV.B     #H'F0,R0L
130 010014 380A 130          MOV.B     R0L,@OUT_DAT+3
131 010016 F8F0 131          MOV.B     #H'F0,R0L
132 010018 380B 132          MOV.B     R0L,@OUT_DAT+4
133 01001A F8F0 133          MOV.B     #H'F0,R0L

```

```

134 01001C 380C          134          MOV.B      R0L,@OUT_DAT+5
135 01001E F8F0          135          MOV.B      #H'F0,R0L
136 010020 380D          136          MOV.B      R0L,@OUT_DAT+6
137 010022 F8F0          137          MOV.B      #H'F0,R0L
138 010024 380E          138          MOV.B      R0L,@OUT_DAT+7
139 010026 F8F0          139          MOV.B      #H'F0,R0L
140 010028 380F          140          MOV.B      R0L,@OUT_DAT+8
141 01002A F8F0          141          MOV.B      #H'F0,R0L
142 01002C 3810          142          MOV.B      R0L,@OUT_DAT+9
143 01002E F8F0          143          MOV.B      #H'F0,R0L
144 010030 3811          144          MOV.B      R0L,@OUT_DAT+10
145 010032 F8F0          145          MOV.B      #H'F0,R0L
146 010034 3812          146          MOV.B      R0L,@OUT_DAT+11
147 010036 F8F0          147          MOV.B      #H'F0,R0L
148 010038 3813          148          MOV.B      R0L,@OUT_DAT+12
149                                149 ;
150                                150 ;***** INITIALIZE ITU *****
151 01003A F820          151          MOV.B      #H'20,R0L
152 01003C 3864          152          MOV.B      R0L,@TCR0 ;Set non overlap time
153 01003E F801          153          MOV.B      #H'01,R0L
154 010040 3866          154          MOV.B      R0L,@TIER0 ;Initialize TIER0(1->IMIEA)
155 010042 7900FF00      155          MOV.W      #H'FF00,R0
156 010046 6B80FF6A      156          MOV.W      R0,@GRA0H ;Initialize TCR0(1->CCLR1)
157 01004A 7F677200      157          BCLR      #0,@TSR0
158                                158 ;
159                                159 ;***** INITIALIZE D/A *****
160 01004E F840          160          MOV.B      #H'40,R0L
161 010050 38DE          161          MOV.B      R0L,@DACR ;Initialize D/AC
162                                162 ;
163                                163 ;***** INITIALIZE DMAC *****
164 010052 7A0000FFFF07  164          MOV.L      #OUT_DAT,ER0 ;Set address of
                                ;transmit data
165 010058 01006B80FFFC  165          MOV.L      ER0,@SIN_ADR
166 01005E 01006901      166          MOV.L      @ER0,ER1 ;load next data bit
167 010062 1031          167          SHLL.L    ER1
168 010064 01006B81FF00  168          MOV.L      ER1,@SIN_DAT
169 01006A 5840000C      169          BCC      SINMN01 ;Branch next data =
                                ;'0'
170                                170 ;
171 01006E 7A000001019A  171          MOV.L      #S12USTADR,ER0 ;Output wave is 1200Hz
172 010074 FA01          172          MOV.B      #H'01,R2L
173 010076 58000008      173          BRA      SINMN02
174                                174 ;
175 01007A 7A00000101C2  175 SINMN01: MOV.L      #S24STADR,ER0 ;Output wave is 2400Hz
176 010080 FA04          176          MOV.B      #H'04,R2L
177                                177 ;

```

```

178 010082 01006B80FF20 178 SINMN02: MOV.L ER0,@MAR0AR
179 010088 3A04 179 MOV.B R2L,@SIN_MOD ;Store next wave type
;to SIN_MOD

180 01008A F8DC 180 MOV.B #H'DC,R0L
181 01008C 3826 181 MOV.B R0L,@IOAR0A ;Set IOAR0A
182 01008E 79000014 182 MOV.W #H'0014,R0
183 010092 6B80FF24 183 MOV.W R0,@ETCR0AH ;Store transfer
;counter

184 010096 2827 184 MOV.B @DTCR0A,R0L ;Load DTCR0A
185 010098 F808 185 MOV.B #H'08,R0L
186 01009A 3827 186 MOV.B R0L,@DTCR0A ;init. DTCR0A
187 01009C 7F277070 187 BSET #DTE,@DTCR0A ;Start DMAC ch0 A
188 188 ;

189 0100A0 F801 189 MOV.B #H'01,R0L ;Initialize counter
190 0100A2 3805 190 MOV.B R0L,@SIN_LWCOUN
191 0100A4 3806 191 MOV.B R0L,@SIN_FCOUNT
192 192 ;

193 0100A6 3860 193 MOV.B R0L,@TSTR ;Timer start
194 0100A8 0700 194 LDC.B #0,CCR ;I flag clr
195 195 ;

196 0100AA 40FE 196 SINMN99: BRA SINMN99
197 197 ;
198 198 ;*****
199 199 ;*
200 200 ;* NAME : SINWAVE(Set next data of sin wave DMAC) *
201 201 ;*
202 202 ;*****
203 203 ;*
204 204 ;* ENTRY : SIN_ADR (Address of output data) *
205 205 ;* ENTRY : SIN_DAT (Next data buff) *
206 206 ;* ENTRY : SIN_MOD (Sin wave type) *
207 207 ;* ENTRY : SIN_LWCOUN (Count of 1 long ward data) *
208 208 ;* ENTRY : SIN_FCOUNT (Count of 100 bits data) *
209 209 ;* RETURNS : NOTHING *
210 210 ;*
211 211 ;*****
212 212 ;

213 000100AC 213 SINWAVE: .EQU $
214 0100AC 01006DF0 214 PUSH.L ER0 ;Escape register
215 0100B0 01006DF1 215 PUSH.L ER1
216 216 ;

217 0100B4 7F677200 217 BCLR.B #0,@TSR0 ;Clear time comparemech
;flag

218 218 ;

219 0100B8 79000014 219 MOV.W #H'0014,R0 ;Set transfer data innning
220 0100BC 6B80FF24 220 MOV.W R0,@ETCR0AH

```

221	0100C0	2827	221	MOV.B	@DTCR0A,R0L	
222			222	;		
223	0100C2	01006B00FF00	223	MOV.L	@SIN_DAT,ER0	;Load next data bit
224	0100C8	1030	224	SHLL.L	ER0	
225	0100CA	01006B80FF00	225	MOV.L	ER0,@SIN_DAT	
226	0100D0	6701	226	BST	#0,R1H	
227			227	;		
228	0100D2	2904	228	MOV.B	@SIN_MOD,R1L	
229	0100D4	A901	229	CMP.B	#H'01,R1L	;SIN_MOD = '1' ?
230	0100D6	5860000A	230	BNE	SINWAVE01	;No
231	0100DA	7301	231	BTST	#0,R1H	;Next data = '1' ?
232	0100DC	5870003C	232	BEQ	SINWAVE05	;Yes
233	0100E0	5800002A	233	BRA	SINWAVE04	;No
234			234	;		
235	0100E4	A902	235	SINWAVE01: CMP.B	#H'02,R1L	;SIN_MOD = '2' ?
236	0100E6	5860000A	236	BNE	SINWAVE02	;No
237	0100EA	7301	237	BTST	#0,R1H	;Next data = '1' ?
238	0100EC	58700048	238	BEQ	SINWAVE07	;Yes
239	0100F0	58000036	239	BRA	SINWAVE06	;No
240			240	;		
241	0100F4	A904	241	SINWAVE02: CMP.B	#H'04,R1L	;SIN_MOD = '4' ?
242	0100F6	5860000A	242	BNE	SINWAVE03	;No
243	0100FA	7301	243	BTST	#0,R1H	;Next data = '1' ?
244	0100FC	58700038	244	BEQ	SINWAVE07	;No
245	010100	58000026	245	BRA	SINWAVE06	;Yes
246			246	;		
247	010104	7301	247	SINWAVE03: BTST	#0,R1H	;Next data = '1' ?
248	010106	58700012	248	BEQ	SINWAVE05	;No
249	01010A	58000000	249	BRA	SINWAVE04	;Yes
250			250	;		
251	01010E	7A00000101AE	251	SINWAVE04: MOV.L	#S12LSTADR,ER0	;Set MAR0AR
252	010114	F908	252	MOV.B	#H'08,R1L	;Set DTCR0A
253	010116	F102	253	MOV.B	#H'02,R1H	;Set SIN_MOD
254	010118	58000026	254	BRA	SINWAVE08	
255			255	;		
256	01011C	7A00000101D5	256	SINWAVE05: MOV.L	#S24EDADR,ER0	;Set MAR0AR
257	010122	F928	257	MOV.B	#H'28,R1L	;Set DTCR0
258	010124	F108	258	MOV.B	#H'08,R1H	;Set SIN_MOD
259	010126	58000018	259	BRA	SINWAVE08	
260			260	;		
261	01012A	7A000001019A	261	SINWAVE06: MOV.L	#S12USTADR,ER0	;Set MAR0AR
262	010130	F908	262	MOV.B	#H'08,R1L	;Set DTCR0A
263	010132	F101	263	MOV.B	#H'01,R1H	;Set SIN_MOD
264	010134	5800000A	264	BRA	SINWAVE08	
265			265	;		
266	010138	7A00000101C2	266	SINWAVE07: MOV.L	#S24STADR,ER0	;Set MAR0AR

```

267 01013E F908          267          MOV.B      #H'08,R1L      ;Set DTCR0A
268 010140 F104          268          MOV.B      #H'04,R1H      ;Set SIN_MOD
269                          269 ;
270 010142 3104          270 SINWAVE08: MOV.B      R1H,@SIN_MOD
271 010144 01006B80FF20  271          MOV.L      ER0,@MAR0AR
272 01014A 3927          272          MOV.B      R1L,@DTCR0A
273 01014C 7F667000      273          BSET.B     #0,@TIER0
274 010150 7F277070      274          BSET.B     #DTE,@DTCR0A
275                          275 ;
276 010154 2805          276          MOV.B      @SIN_LWCOUN,R0L ;Increment counter
277 010156 0A08          277          INC.B      R0L
278 010158 3805          278          MOV.B      R0L,@SIN_LWCOUN
279                          279 ;
280 01015A A820          280          CMP.B      #32,R0L      ;32bit out end ?
281 01015C 58600020      281          BNE       SINWAVE09    ;No
282 010160 F800          282          MOV.B      #0,R0L      ;Clear counter
283 010162 3805          283          MOV.B      R0L,@SIN_LWCOUN
284                          284 ;
285 010164 01006B00FFFC  285          MOV.L      @SIN_ADR,ER0 ;Set next data address
286 01016A 7A1000000004  286          ADD.L      #4,ER0
287 010170 01006B80FEFC  287          MOV.L      ER0,@SIN_ADR
288                          288 ;
289 010176 01006900      289          MOV.L      @ER0,ER0    ;Set next data
290 01017A 01006B80FF00  290          MOV.L      ER0,@SIN_DAT
291                          291 ;
292 010180 2806          292 SINWAVE09: MOV.B     @SIN_FCOUNT,R0L ;Increment counter
293 010182 0A08          293          INC.B      R0L
294 010184 3806          294          MOV.B      R0L,@SIN_FCOUNT
295                          295 ;
296 010186 A864          296          CMP.B      #100,R0L    ;100bit out end ?
297 010188 58600004      297          BNE       SINWAVE99    ;No
298                          298 ;
299 01018C F800          299          MOV.B      #0,R0L      ;Yes,Stop timer
300 01018E 3860          300          MOV.B      R0L,@TSTR
301                          301 ;
302 010190 01006D71      302 SINWAVE99: POP.L     ER1
303 010194 01006D70      303          POP.L     ER0
304 010198 5670          304          RTE
305                          305 ;
306                          306 ;*****
307                          307 ;*
308                          308 ;*      DATA TABLE (OUTPUT DATA TAB)
309                          309 ;*
310                          310 ;*****
311                          311 ;
312 01019A 8093A7B9CA      312 S12USTADR      .DATA.B H'80,H'93,H'A7,H'B9,H'CA

```


313	01019F	DAE7F1F9FD	313	.DATA.B H'DA,H'E7,H'F1,H'F9,H'FD
314	0101A4	FFFDF9F1E7	314	.DATA.B H'FF,H'FD,H'F9,H'F1,H'E7
315	0101A9	DACAB9A793	315	.DATA.B H'DA,H'CA,H'B9,H'A7,H'93
316	0101AE	806C584635	316	S12LSTADR .DATA.B H'80,H'6C,H'58,H'46,H'35
317	0101B3	25180E0602	317	.DATA.B H'25,H'18,H'0E,H'06,H'02
318	0101B8	0002060E18	318	.DATA.B H'00,H'02,H'06,H'0E,H'18
319	0101BD	253546586C	319	.DATA.B H'25,H'35,H'46,H'58,H'6C
320	0101C2	80A7CAE7F9	320	S24STADR .DATA.B H'80,H'A7,H'CA,H'E7,H'F9
321	0101C7	FFF9E7CAA7	321	.DATA.B H'FF,H'F9,H'E7,H'CA,H'A7
322	0101CC	8058351806	322	.DATA.B H'80,H'58,H'35,H'18,H'06
323	0101D1	00061835	323	.DATA.B H'00,H'06,H'18,H'35
324	0101D5	58	324	S24EDADR .DATA.B H'58
325			325	;
326			326	.END

*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0

3.4 Four-Phase Stepping Motor Application Example

MCU: H8/3003

Function: ITU, TPC, DMAC

3.4.1 Specifications

- Control of 4-phase stepping motor using H8/3003 on-chip functions of ITU, TPC and DMAC (see figure 3.22)
- Stepping motor control by 2-phase drive system
- This sample task repeats the operations “stop -> forward -> stop -> reverse -> stop”
- This sample task does not require software intervention to conduct slue-up and slue-down processing
- A through-current prevention interval is introduced to protect the motor driver

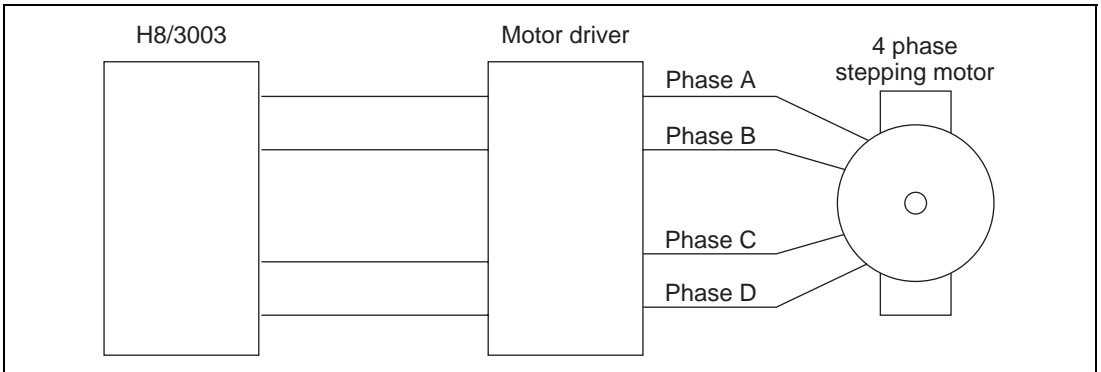


Figure 3.22 4-Phase Stepping Motor Control Circuit

3.4.2 Method

Example of Stepping Motor Operation: Figure 3.23 shows how 2-phase drive is employed to operate a 4-phase stepping motor. The following is an overview of operations:

- As shown in figure 3.23, when the pulse is high, the corresponding phase is driven.
- In (1), phase A and phase D undergo simultaneous drive. At this time the rotor is at an intermediate position between phase D and phase A.
- In (2), phase A and phase B undergo simultaneous drive. At this time the rotor is between phase A and phase B. The 2-phase drive system then excites the neighboring two phases in sequence to turn the rotor (A-D → A-B → B-C → C-D).
- In reverse operation, the sequence of excitation is reverse (D-C → C-B → B-A → A-D).
- Stopping the motor is achieved by excitation for a constant period at the last phase of forward or reverse rotation.

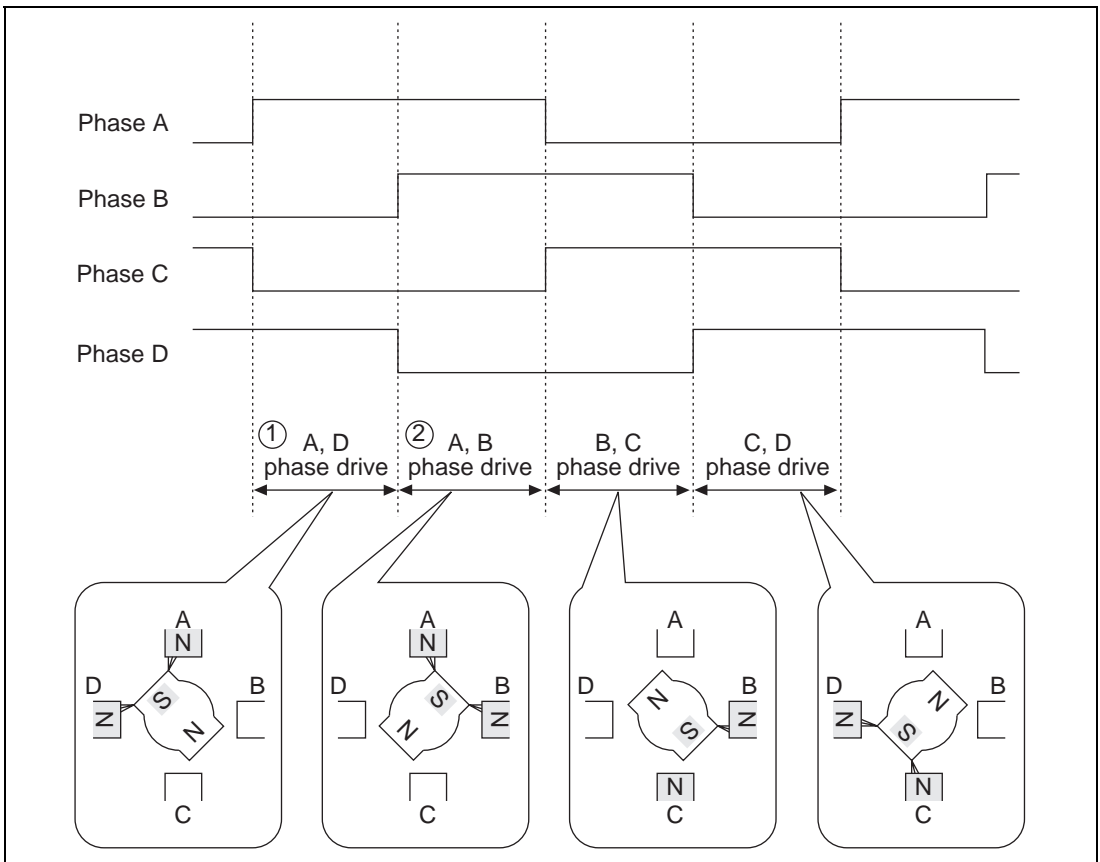


Figure 3.23 Example of Stepping Motor Operation

Non-Overlap Time: As shown in figure 3.24, a through-current prevention period (n) or non-overlap time is inserted during output pattern switching. Owing to turn-off lag during a drive phase switch, there is a danger of the driver being destroyed. Non-overlap time provides a time delay to counter this.

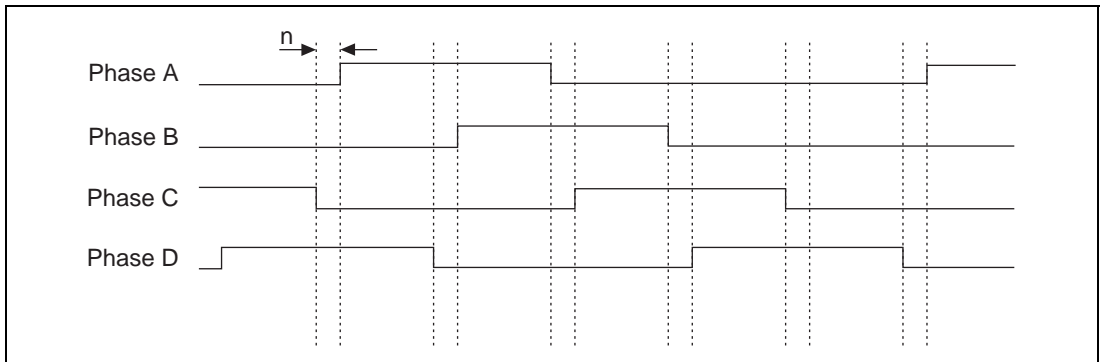


Figure 3.24 Example of Non-Overlap Time Output

Slue-Up and Slue-Down Operations: As shown in figure 3.25, acceleration/deceleration-controlled pulses are output. Conducting slue-up and slue-down operation in this way prevents step-out of the motor.

When the motor is running, if there is a sudden output of short pulses, the motor may be unable to take up the load and fail to rotate. Slue-up and slue-down operation is a means of preventing this.

The principle is as follows:

1. Gradual shortening of pulse cycle and set amount of pulse output (slue-up)
2. Preset amount of pulse output with a constant pulse cycle (constant speed)
3. Gradual lengthening of pulse cycle and set amount of pulse output (slue-down)

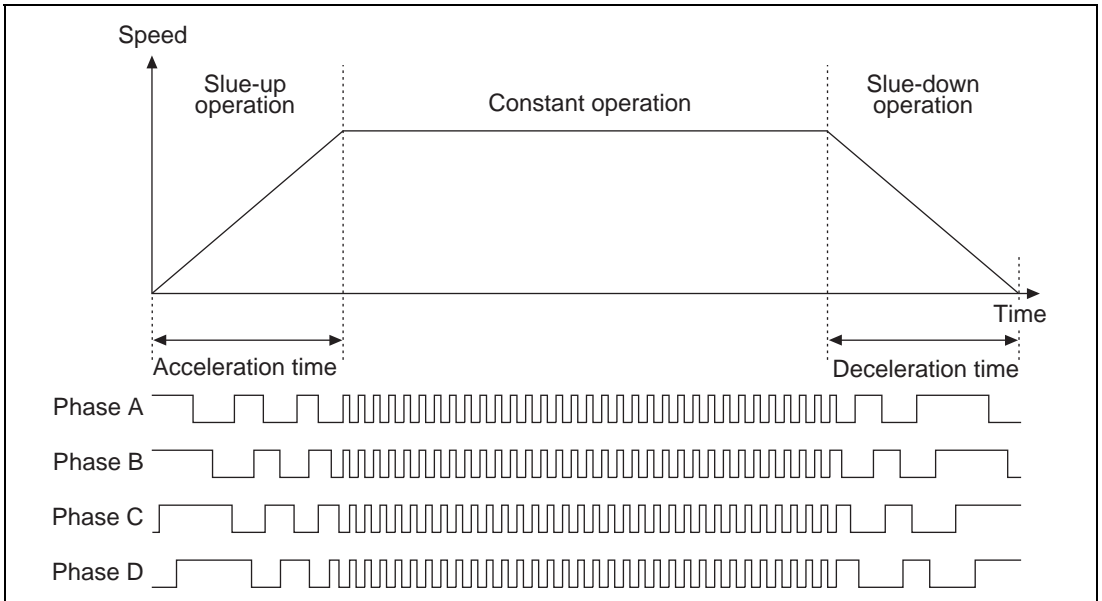


Figure 3.25 Slue-Up and Slue-Down Operation

3.4.3 Description of Functions Used

Figure 3.26 shows the H8/3003 on-chip functions used in this sample task.

Two DMAC channels are used, and a 4-phase output pattern is subjected to timing pattern control (TPC) to give a 4-phase pulse output. Also, the pulse cycle is transferred to the ITU's GRB to control 4-phase pulse output timing control.

Cycle Data Table: Table for altering the step cycle (ITU timer values) set in ROM.

Output Pattern Data Table: Output pattern set in ROM for 4-phase pulse output.

DMAC0A: Starts up by ITU compare/match A. Transfers output pattern from output pattern data table to the TPC's NDR.

DMAC0B: Starts up by ITU compare/match A. Transfers cycle data from cycle data table to the ITU's GRB.

ITU:

- Compare/match A: Starts up DMAC0A, DMAC0B and TPC
- Compare/match B: Clears timer counter and starts up TPC

TPC: Uses the DMAC to effect data output without CPU intervention. Although a maximum of 16-bit data can be output, this sample task outputs 4 bit data. Enables output of pulses with non-overlap time.

- Compare/match B: Outputs pulses changing from high to low. Output changing from low to high is held.
- Compare/match A: Conducts output of pulse held by compare/match B (L -> H pulse). (Delayed by GRA set value.)

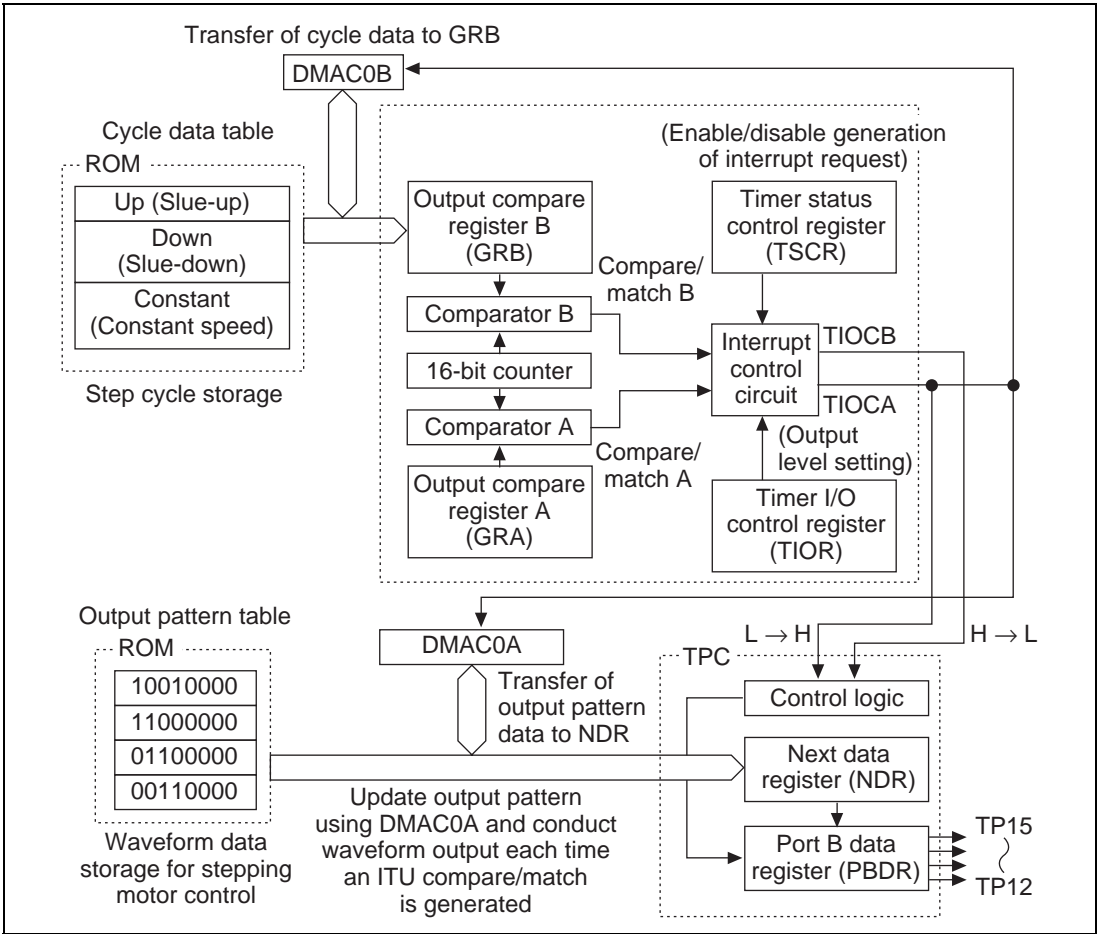


Figure 3.26 Block Diagram of 4-Phase Stepping Motor Control

Table 3.19 shows the function allocation for this sample task. H8/3003 functions are allocated to generate 2-phase drive stepping motor output waveforms.

Table 3.19 H8/3003 Function Allocation

H8/3003 Function	Function	
ITU (ch0)	TCNT0	16-bit counter
	GRA0	Output compare register A
	GRB0	Output compare register B
	TCR0	Selects count clock and sets counter clear factor
	TIOR0	Sets GRA and GRB to output compare registers
	TIER0	Controls interrupt request enable/disable
	TSR0	Compare/match and overflow status register
	TSTR	Enables/disables TCNT counting operation
DMAC0A/B	DTCR0A, B	Controls operation of each DMAC channel
	MAR0A, B	Sets transfer source address
	IORA0A, B	Sets transfer destination address
	ETCR0A, B	Sets number of transfers
TPC	PBDDR	Sets TPC output pins
	PBDR	Stores TPC output B data
	TPMR	Sets TPC output non-overlap mode
	TPCR	Sets TPC output trigger signal
	NDERB	Enables/disables TPC output B
	NDRB	Stores next TPC output B data
	TP15 – TP12	Generates 4-phase stepping motor output waveforms

3.4.4 Description of Operations

4-phase pulse output (with non-overlap time) (figure 3.27).

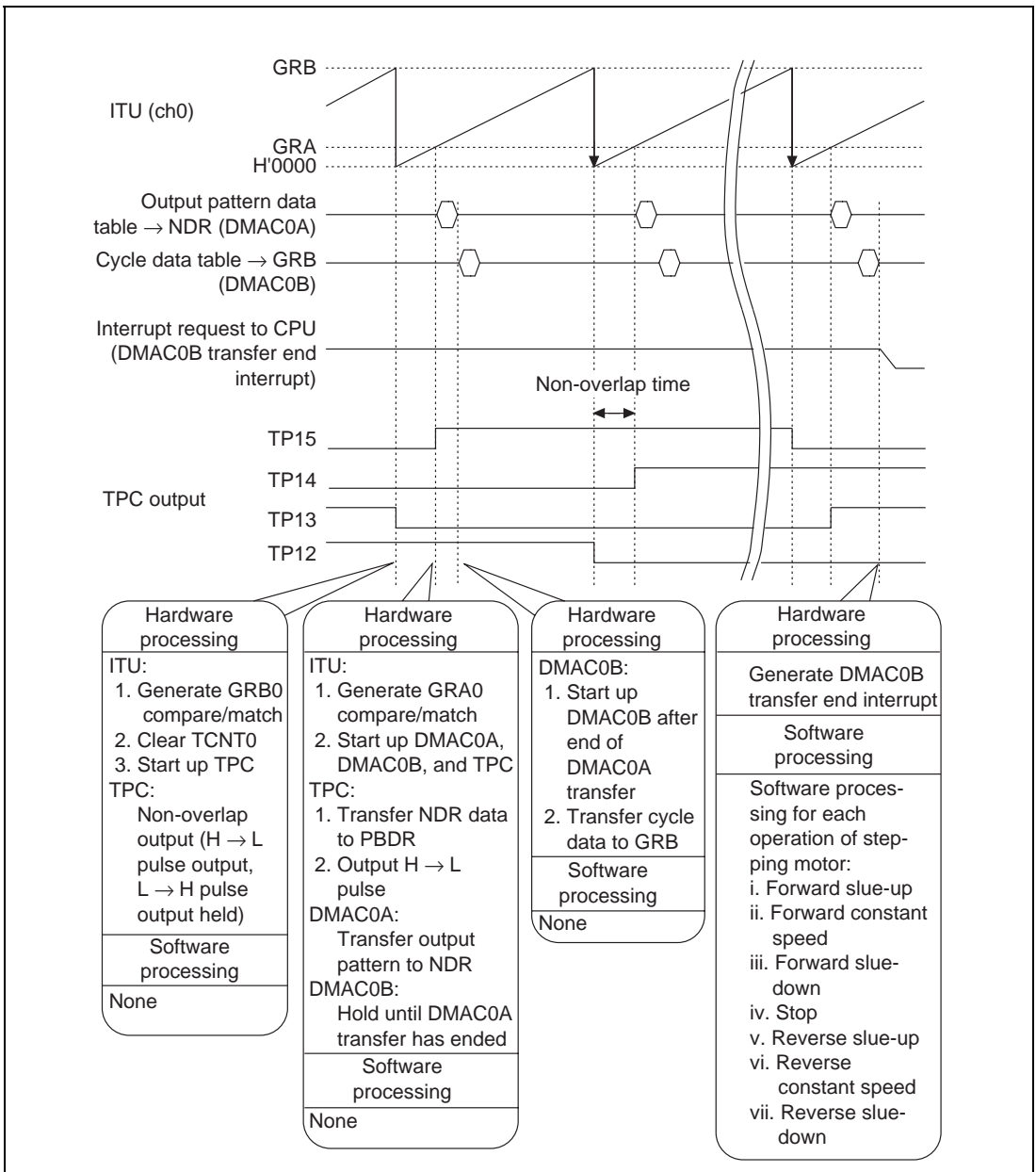


Figure 3.27 Stepping Motor Timing Description

3.4.5 Description of Software

The software for the four-phase stepping motor application example is described below.

Table 3.20 Description of Modules

Module Name	Label Name	Function
Main routine	STP4MN	ITU, TPC and DMAC0A/DMAC0B initial settings, and forward slue-up operation
Motor control	STP41	Motor operations (forward constant speed, stop etc.) control <ul style="list-style-type: none">• Sets DMAC0A/B and conducts forward slue-up operation• Sets DMAC0B and conducts forward constant speed operation• Sets DMAC0A/B and conducts forward slue-down operation• Sets DMAC0A/B and conducts stopping operation• Sets DMAC0A/B and conducts reverse slue-up operation• Sets DMAC0B and conducts reverse constant speed operation• Sets DMAC0A/B and conducts reverse slue-down operation• Sets DMAC0A/B and conducts stopping operation

Table 3.21 Description of Internal Registers

Onchip Function	Register Name	Function
ITU	GRA0	Sets non-overlap time
	GRB0	Sets timer cycle
	TIER0	Enables IMFA interrupts
	TCR0	Sets ITU as follows: <ul style="list-style-type: none"> • Clear counter with GRB compare/match • Count by internal clock ϕ
	TIOR0	Sets GRA and GRB to output compare registers, and prohibits pin output
	TSTR	Enables TCNT0 counting operation
DMAC0A	DTCR0A	Sets DMAC0A as follows: <ul style="list-style-type: none"> • Data size to byte size • MAR to increment • Data transfer to repeat mode • Transfer enable or disable selection (differs depending on the module) • Startup factor to ITU ch0 compare/match A
	MAR0A	Sets output pattern data table transfer source address
	IOAR0A	Sets NDRB address (transfer destination)
	ETCR0A	Sets number of transfers
	DMAC0B	DTCR0B
	MAR0B	Sets cycle data table transfer source address
	IOAR0B	Sets GRB address (transfer destination)
	ETCR0B	Sets number of transfers

Table 3.21 Description of Internal Registers (cont)

Onchip Function	Register Name	Function
TPC	PBDDR	Enables TP15- TP12 TPC output
	PBDR	Stores output pattern data
	TPMR	Sets TP15- TP12 to non-overlap output
	TPCR	Sets TP15- TP12 output trigger to ITU ch0 compare/match
	NDERB	Enables TP15- TP12 TPC output
	NDRB	Stores next output pattern data

Description of Arguments: Arguments are not used in this sample task.

Table 3.22 Description of RAM

Label Name	Function	Data Length	Module Name
CTRLMD	Indicates operation mode of 4-phase stepping motor.	1 byte	Motor control
	H'01: forward slue-up operation end		
	H'02: forward constant speed operation end		
	H'03: forward slue-down operation end		
	H'00: forward stop operation end		
	H'80: reverse stop operation end		
	H'81: reverse slue-up operation end		
	H'82: reverse constant speed operation end		
H'83: reverse slue-down operation end			

Table 3.23 Description of Data Tables

Table Name	Function	Data Length	Data Volume
PATTBL	Sets pattern for 4-phase pulse output	Byte	4 bytes
UPTBL	Sets data for changing step cycle	Word	121 words
DOWNTBL			
CNSTBL			

Description of General Registers: Figure 3.28 describes the general registers.

	31	ERn	RnH	RnL	0
0	Used as work register during data setting in all routines				
1					
2					
3					
4					
5					
6					
7					

Figure 3.28 Description of General Registers

3.4.6 Flowcharts

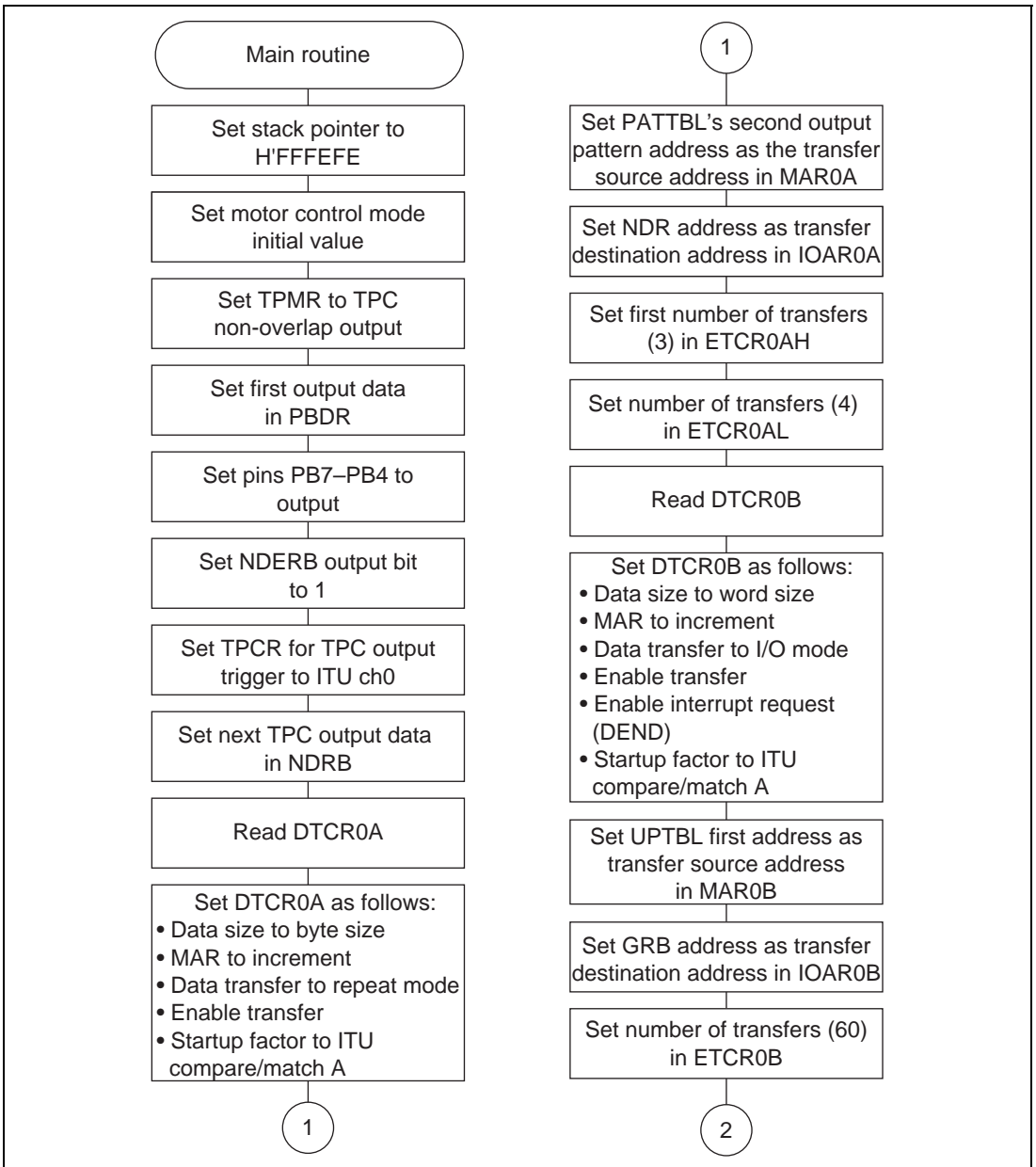


Figure 3.29 Main Routine Flowchart (1)

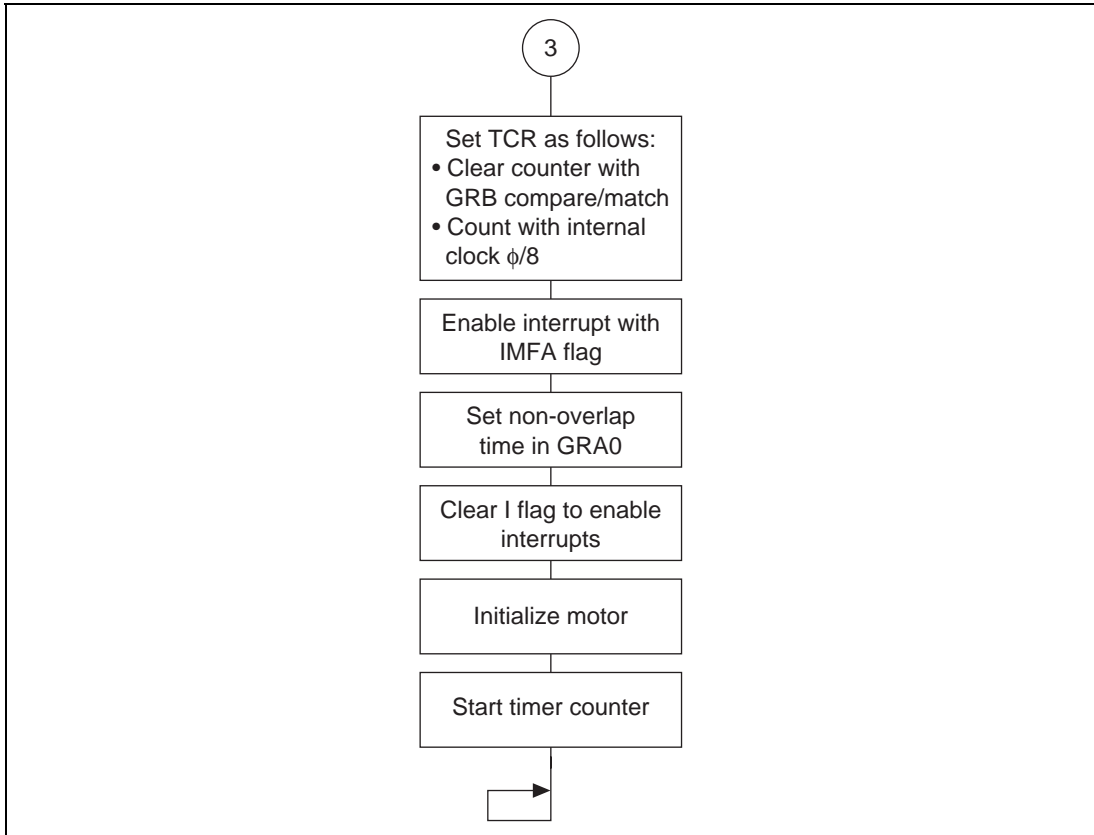


Figure 3.29 Main Routine Flowchart (2)

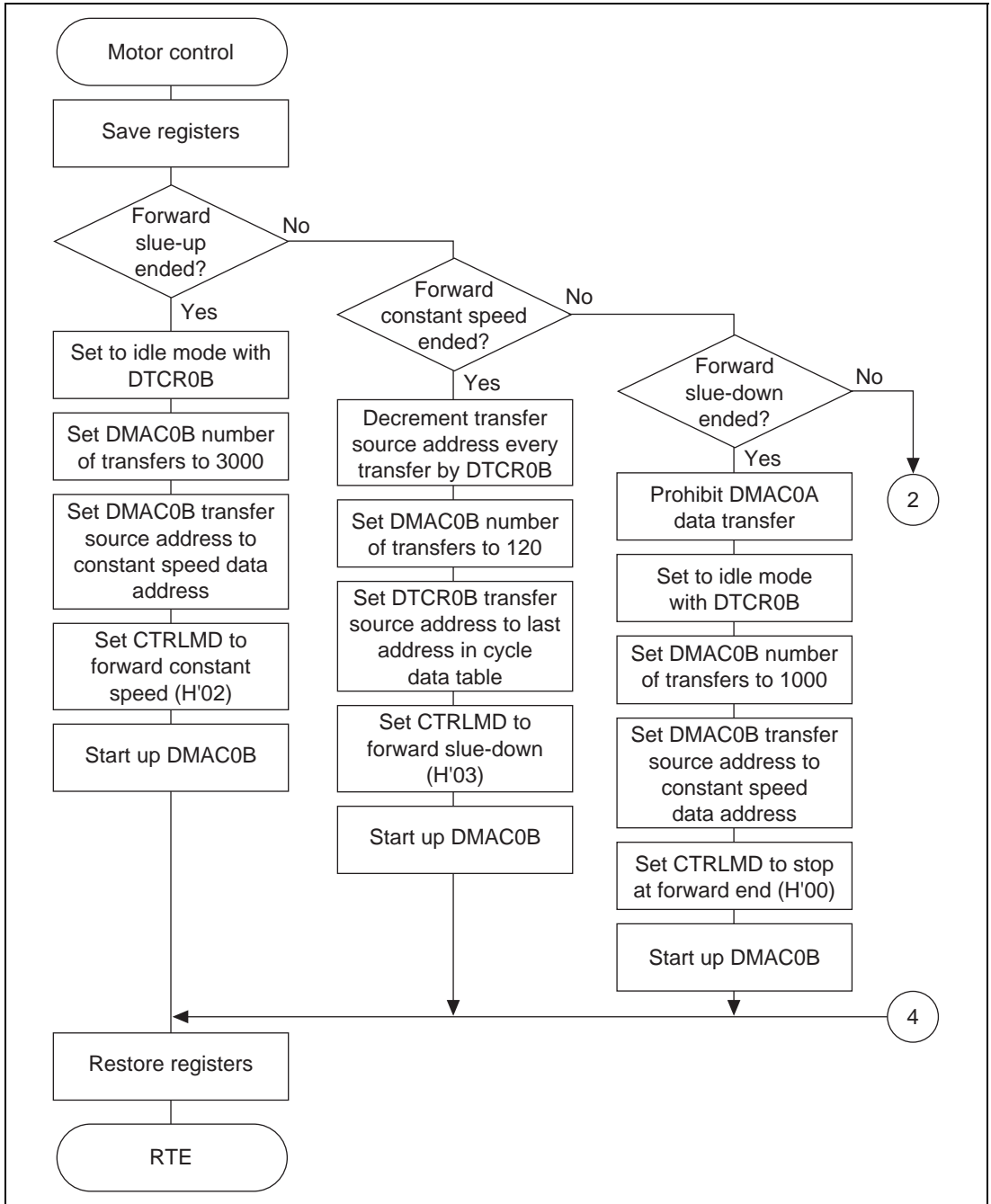


Figure 3.30 Motor Control Flowchart (1)

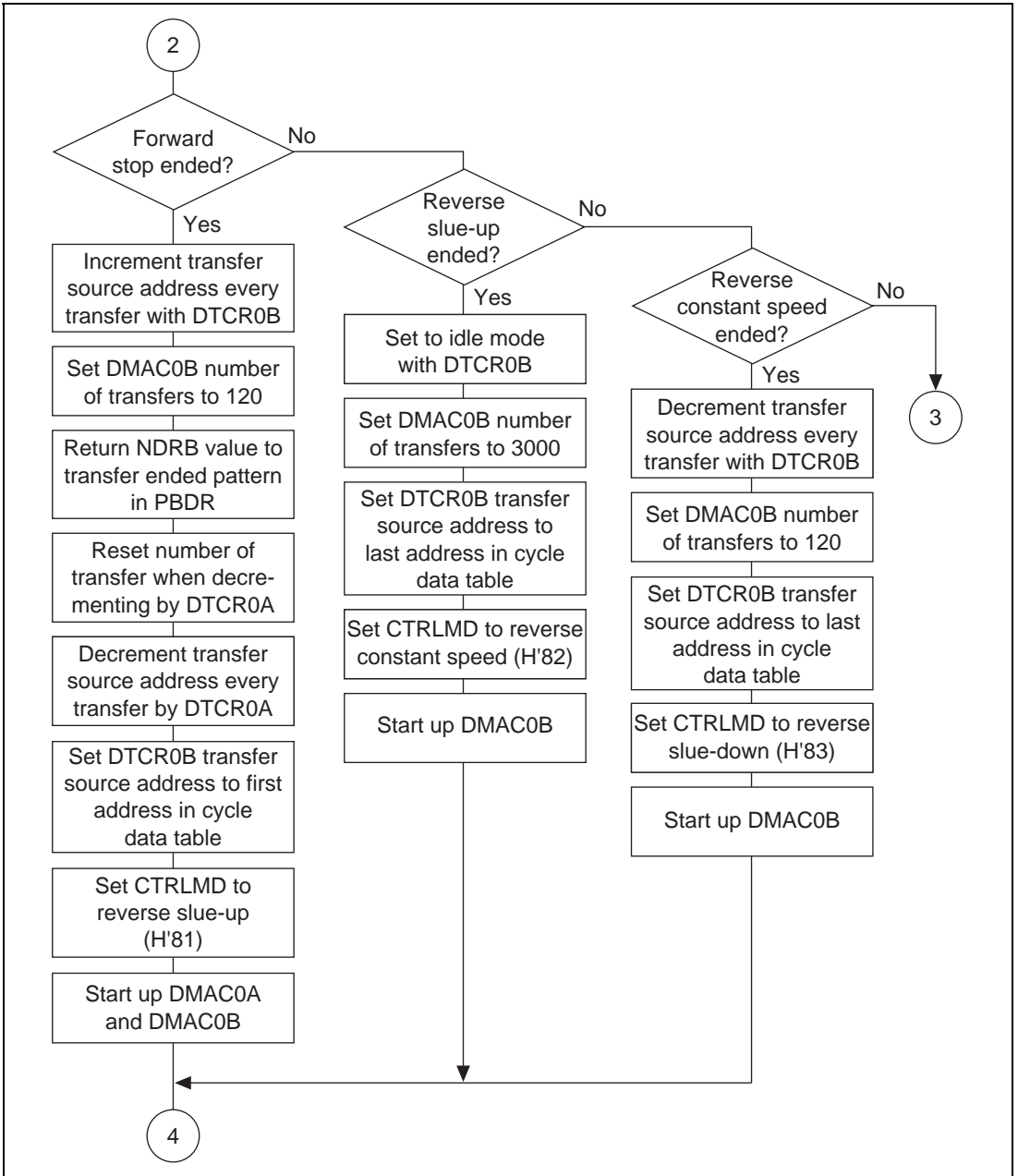


Figure 3.30 Motor Control Flowchart (2)

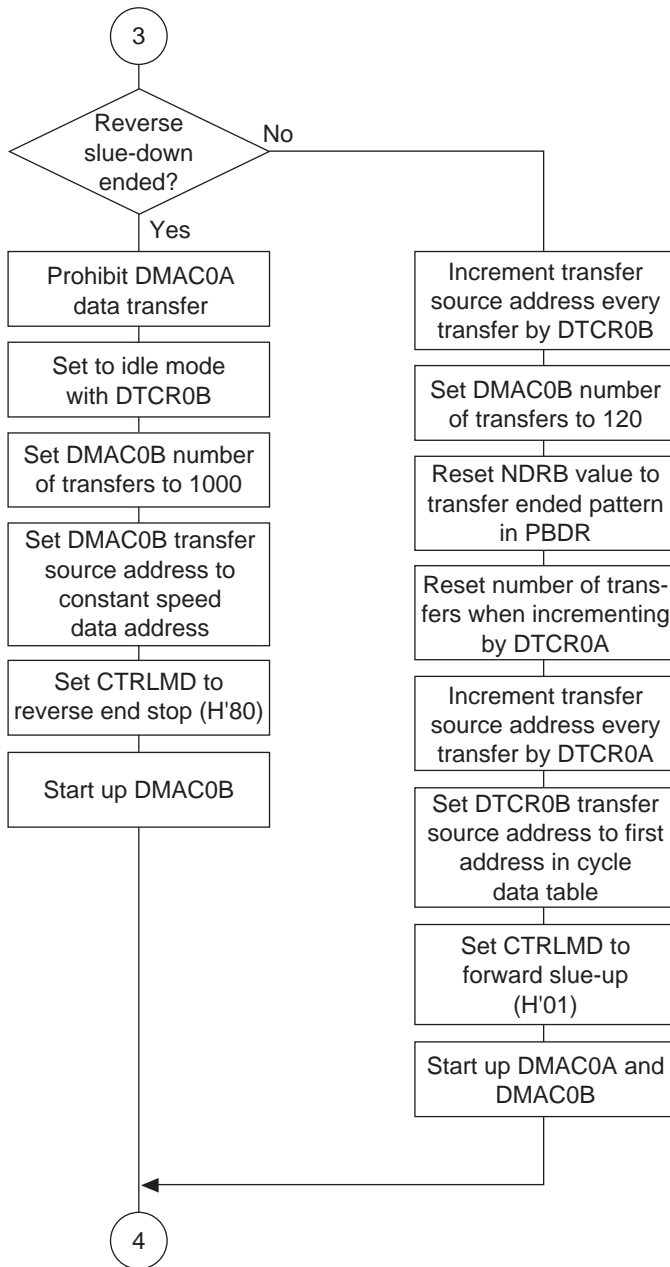


Figure 3.30 Motor Control Flowchart (3)

3.4.7 Program List

```

1 ;*****
2 ;*
3 ;* VECTOR ADDRESS *
4 ;*
5 ;*****
6 ;
7 000000 7 .SECTION VECT, CODE, LOCATE=H'000000
8 ;
9 000000 00010000 9 RES .DATA.L STP4MN
10 ;
11 00001C 11 .ORG H'00001C
12 00001C 00010000 12 NMI .DATA.L STP4MN
13 000020 00010000 13 TRAPA0 .DATA.L STP4MN
14 000024 00010000 14 TRAPA1 .DATA.L STP4MN
15 000028 00010000 15 TRAPA2 .DATA.L STP4MN
16 00002C 00010000 16 TRAPA3 .DATA.L STP4MN
17 ;
18 000030 18 .ORG H'000030
19 000030 00010000 19 IRQ0 .DATA.L STP4MN
20 000034 00010000 20 IRQ1 .DATA.L STP4MN
21 000038 00010000 21 IRQ2 .DATA.L STP4MN
22 00003C 00010000 22 IRQ3 .DATA.L STP4MN
23 000040 00010000 23 IRQ4 .DATA.L STP4MN
24 000044 00010000 24 IRQ5 .DATA.L STP4MN
25 000048 00010000 25 IRQ6 .DATA.L STP4MN
26 00004C 00010000 26 IRQ7 .DATA.L STP4MN
27 000050 00010000 27 WOVI0 .DATA.L STP4MN
28 000054 00010000 28 CMI .DATA.L STP4MN
29 ;
30 000060 30 .ORG H'000060
31 000060 00010000 31 IMIA0 .DATA.L STP4MN
32 000064 00010000 32 IMIB0 .DATA.L STP4MN
33 000068 00010000 33 OVI0 .DATA.L STP4MN
34 ;
35 000070 35 .ORG H'000070
36 000070 00010000 36 IMIA1 .DATA.L STP4MN
37 000074 00010000 37 IMIB1 .DATA.L STP4MN
38 000078 00010000 38 OVI1 .DATA.L STP4MN
39 ;
40 000080 40 .ORG H'000080
41 000080 00010000 41 IMIA2 .DATA.L STP4MN
42 000084 00010000 42 IMIB2 .DATA.L STP4MN
43 000088 00010000 43 OVI2 .DATA.L STP4MN

```

```

44 ;
45 000090 45 .ORG H'000090
46 000090 00010000 46 IMIA3 .DATA.L STP4MN
47 000094 00010000 47 IMIB3 .DATA.L STP4MN
48 000098 00010000 48 OVI3 .DATA.L STP4MN
49 ;
50 0000A0 50 .ORG H'0000A0
51 0000A0 00010000 51 IMIA4 .DATA.L STP4MN
52 0000A4 00010000 52 IMIB4 .DATA.L STP4MN
53 0000A8 00010000 53 OVI4 .DATA.L STP4MN
54 ;
55 0000B0 55 .ORG H'0000B0
56 0000B0 00010000 56 DEND0A .DATA.L STP4MN
57 0000B4 0001007C 57 DEND0B .DATA.L STP41
58 0000B8 00010000 58 DEND1A .DATA.L STP4MN
59 0000BC 00010000 59 DEND1B .DATA.L STP4MN
60 0000C0 00010000 60 DEND2A .DATA.L STP4MN
61 0000C4 00010000 61 DEND2B .DATA.L STP4MN
62 0000C8 00010000 62 DEND3A .DATA.L STP4MN
63 0000CC 00010000 63 DEND3B .DATA.L STP4MN
64 0000D0 00010000 64 ERI0 .DATA.L STP4MN
65 0000D4 00010000 65 RXI0 .DATA.L STP4MN
66 0000D8 00010000 66 TXI0 .DATA.L STP4MN
67 0000DC 00010000 67 TEI0 .DATA.L STP4MN
68 0000E0 00010000 68 ERI1 .DATA.L STP4MN
69 0000E4 00010000 69 RXI1 .DATA.L STP4MN
70 0000E8 00010000 70 TXI1 .DATA.L STP4MN
71 0000EC 00010000 71 TEI1 .DATA.L STP4MN
72 0000F0 00010000 72 ADI .DATA.L STP4MN
73 ;
74 ;*****
75 ;* *
76 ;* RAM ALLOCATION *
77 ;* *
78 ;*****
79 ;
80 FFFF00 80 .SECTION RAM,DATA,LOCATE=H'FFFF00
81 ;
82 FFFF00 00000001 82 CTRLMD .RES.B 1 ;Motor control mode
83 ;
84 ;*****
85 ;* *
86 ;* SYMBOL DEFINITIONS *
87 ;* *
88 ;*****
89 ;

```

```

90 ;***** DMAC0A *****
91 ;
92 00FFFF21 MAR0A .EQU H'FFFF21 ;Memory address register 0A
93 00FFFF24 ETCR0AH .EQU H'FFFF24 ;Execute transfer control
;register 0AH
94 00FFFF25 ETCR0AL .EQU H'FFFF25 ;Execute transfer control
;register 0AL
95 00FFFF26 IOAR0A .EQU H'FFFF26 ;I/O address register 0A
96 00FFFF27 DTCR0A .EQU H'FFFF27 ;Data transfer control
;register 0A
97 ;
98 ;***** DMAC0B *****
99 ;
100 00FFFF29 MAR0B .EQU H'FFFF29 ;Memory address register 0B
101 00FFFF2C ETCR0B .EQU H'FFFF2C ;Execute transfer control
;register 0B
102 00FFFF2E IOAR0B .EQU H'FFFF2E ;I/O address register 0B
103 00FFFF2F DTCR0B .EQU H'FFFF2F ;Data transfer control
;register
104 ;
105 ;***** ITU ch0 *****
106 ;
107 00FFFF60 TSTR .EQU H'FFFF60 ;Timer start register
108 00FFFF64 TCR0 .EQU H'FFFF64 ;Timer control register0
109 00FFFF66 TIER0 .EQU H'FFFF66 ;Timer interrupt enable
;register0
110 00FFFF6A GRA0 .EQU H'FFFF6A ;General register A0
111 000006A GRA0L .EQU H'6A
112 00FFFF6C GRB0 .EQU H'FFFF6C ;General register B0
113 000006C GRB0L .EQU H'6C
114 ;
115 ;***** TPC *****
116 ;
117 00FFFA0 TPMR .EQU H'FFFA0 ;TPC output mode register
118 00FFFA1 TPCR .EQU H'FFFA1 ;TPC output control register
119 00FFFA2 NDERB .EQU H'FFFA2 ;Next data enable register B
120 00000A4 NDRBL .EQU H'A4
121 00FFFA4 NDRB .EQU H'FFFA4 ;Next data register B
122 ;
123 ;***** PORT B *****
124 ;
125 00FFFD6 PBDR .EQU H'FFFD6 ;PortB data register
126 00FFFD4 PBDDR .EQU H'FFFD4 ;PortB data direction
;register
127 ;

```

```

128 ;*****
129 ;*
130 ;*      MAIN PROGRAM : STP4MN
131 ;*
132 ;*****
133 ;
134 010000      134      .SECTION      PROG, CODE, LOCATE=H'010000
135
136      00010000      136 STP4MN      .EQU      $
137 010000      7A0700FFFEFE      137      MOV.L      #H'FFFEFE, SP      ;Initialize stack pointer
138 010006      F801      138      MOV.B      #H'01,R0L      ;Initialize mdoe counter
139 010008      3800      139      MOV.B      R0L,@CTRLMD
140
141      141 ;***** INITIALIZE TPC *****
142 ;
143 01000A      F8F8      143      MOV.B      #B'11111000,R0L
144 01000C      38A0      144      MOV.B      R0L,@TPMR      ;Initialize TPMR
                                ;(TP15-12 nonoverlap)
145 01000E      F890      145      MOV.B      #B'10010000,R0L
146 010010      38D6      146      MOV.B      R0L,@PBDR      ;Initialize PBDR
147 010012      F8F0      147      MOV.B      #B'11110000,R0L
148 010014      38D4      148      MOV.B      R0L,@PBDDR      ;Initialize PBDDR
                                ;(B4-B7 output)
149 010016      38A2      149      MOV.B      R0L,@NDERB      ;Initialize NDERB
                                ;(TP15-12 enable)
150 010018      F83F      150      MOV.B      #B'00111111,R0L
151 01001A      38A1      151      MOV.B      R0L,@TPCR      ;Initialize TPCR
                                ;(Select ITU0 trigger)
152 01001C      F090      152      MOV.B      #B'10010000,ROH
153 01001E      30A4      153      MOV.B      ROH,@NDRB      ;Initialize NDRB
154 ;
155 ;***** INITIALIZE DMAC0A *****
156 ;
157 010020      2827      157      MOV.B      @DTCR0A,R0L      ;Dummy read
158 010022      3027      158      MOV.B      R0H,@DTCR0A      ;Initialize DTCR0A
                                ;(repeat mode)
159 010024      7A000001035B      159      MOV.L      #PATTBL+1,ER0;Initialize MAR0A
160 01002A      01006B80FF21      160      MOV.L      ER0,@MAR0A
161 010030      F0A4      161      MOV.B      #NDRBL,R0H      ;Initialize IOAR0A
162 010032      3026      162      MOV.B      R0H,@IOAR0A
163 010034      F803      163      MOV.B      #H'03,R0L      ;Store first transfer
                                ;count
164 010036      3824      164      MOV.B      R0L,@ETCR0AH
165 010038      F804      165      MOV.B      #04,R0L      ;Store transfer count
166 01003A      3825      166      MOV.B      R0L,@ETCR0AL
167 ;

```

```

168 ;***** INITIALIZE DMACOB *****
169 ;
170 01003C 202F 170 MOV.B @DTCR0B,R0H ;Dummy read
171 01003E F0C8 171 MOV.B #B'11001000,R0H ;Initialize DTCR0B
; (I/O mode)
172 010040 302F 172 MOV.B R0H,@DTCR0B
173 010042 7A0000010268 173 MOV.L #UPTL,ER0 ;Initialize MAR0B
174 010048 01006B80FF29 174 MOV.L ER0,@MAR0B
175 01004E F86C 175 MOV.B #GRB0L,R0L ;Initialize IOAR0B
176 010050 382E 176 MOV.B R0L,@IOAR0B
177 010052 7900003C 177 MOV.W #D'60,R0 ;Store transfer count
178 010056 6B80FF2C 178 MOV.W R0,@ETCR0B
179 ;
180 ;***** INITIALIZE ITU *****
181 ;
182 01005A F8C3 182 MOV.B #B'11000011,R0L
183 01005C 3864 183 MOV.B R0L,@TCR0 ;Initialize TCR0 (GRB
; timer clear0)
184 01005E F0F9 184 MOV.B @B'11111001,R0H
185 010060 3066 185 MOV.B R0H,@TIER0 ;Initialize TIER0
; (Interport enemie)
186 010062 7900000A 186 MOV.W #D'10,R0 ;Initialize GRA0
187 010066 6B80FF6A 187 MOV.W R0,@GRA0
188 ;
189 ;***** ENABLE INTERRUPT *****
190 ;
191 01006A 0700 191 LDC.B #B'00000000,CCR
192 ;
193 ;***** START ITU *****
194 ;
195 01006C 7A0000145855 195 MOV.L #H'145855,ER0 ;Initialize motor
196 010072 1B70 196 STP4MN1 DEC.L #1,ER0
197 010074 46FC 197 BNE STP4MN1
198 010076 F8E1 198 MOV.B #B'11100001,R0L
199 010078 3860 199 MOV.B R0L,@TSTR ;Set TCNT0 start flag
200 01007A 40FE 200 STP4MN99 BRA STP4MN99
201 ;
202 ;*****
203 ;* *
204 ;* NAME : STP41(MOTOR CONTROL) *
205 ;* *
206 ;*****
207 ;* *
208 ;* ENTRY : NOTHING *
209 ;* RETURNS : NOTHING *
210 ;* *

```

```

211 ;*****
212 ;
213 STP41 .EQU $
214 01007C 01006DF0 214 PUSH.L ERO
215 010080 2800 215 MOV.B @CTRLMD,R0L
216 ;
217 010082 A801 217 CMP.B #H'01,R0L ;Branch if slue up(CW)
;completed
218 010084 5870002A 218 BEQ NEXT_C_CW
219 ;
220 010088 A802 220 CMP.B #H'02,R0L ;Branch if constant(CW)
;completed
221 01008A 58700048 221 BEQ NEXT_D_CW
222 ;
223 01008E A803 223 CMP.B #H'03,R0L ;Branch if slue down(CW)
;completed
224 010090 58700066 224 BEQ NEXT_S_CW
225 ;
226 010094 A800 226 CMP.B #H'0,R0L ;Branch if stop completed
227 010096 58700088 227 BEQ NEXT_U_CCW
228 ;
229 01009A A881 229 CMP.B #H'81,R0L ;Branch if slue up(CCW)
;completed
230 01009C 587000EA 230 BEQ NEXT_C_CCW
231 ;
232 0100A0 A882 232 CMP.B #H'82,R0L ;Branch if constant(CCW)
;completed
233 0100A2 58700108 233 BEQ NEXT_D_CCW
234 ;
235 0100A6 A883 235 CMP.B #H'83,R0L ;Branch if slue down(CCW)
;completed
236 0100A8 58700126 236 BEQ NEXT_S_CCW
237 ;
238 0100AC A880 238 CMP.B #H'00,R0L ;Branch if stop completed
239 0100AE 58700148 239 BEQ NEXT_U_CW
240 ;
241 ;***** NEXT CONSTANT CW *****
242 ;
243 0100B2 F858 243 NEXT_C_CW MOV.B #B'01011000,R0L
244 0100B4 382F 244 MOV.B R0L,@DTCR0B ;Store report mode data
;to DTCR0B
245 0100B6 79000BB8 245 MOV.W #D'3000,R0 ;Store transfer count to
;ETCR0B
246 0100BA 6B80FF2C 246 MOV.W R0,0ETCR0B
247 0100BE 7A0000010358 247 MOV.L #CNSTBL,ERO ;Store data address to
;MAR0B

```


248	0100C4	01006B60FF29	248	MOV.L	ER0,@MAR0B	
249	0100CA	F802	249	MOV.B	#'02,R0L	;Set mode flag
250	0100CC	3800	250	MOV.B	R0L,@CTRLMD	
251	0100CE	7F2F7070	251	BSET	#7,@DTCR0B	;DMAC channel 1 start
252	0100D2	5800018C	252	BRA	EXIT	
253			253			;
254			254			;***** NEXT SLUJ DOWN CW *****
255			255			;
256	0100D6	F868	256	NEXT_D_CW	MOV.B	#B'01101000,R0L ;Set MAR0B decrement
257	0100D8	382F	257		MOV.B	R0L,@DTCR0B
258	0100DA	79000078	258		MOV.W	#D'120,R0 ;Store transfer count ;to ETCR0B
259	0100DE	6B80FF2C	259		MOV.W	R0,@ETCR0B
260	0100E2	7A0000010356	260		MOV.L	#DOWNTEL,ER0 ;Store data address to ;MAR0B
261	0100E8	01006B80FF29	261		MOV.L	ER0,@MAR0B
262	0100EE	F803	262		MOV.B	##'03,R0L ;Set mode flag
263	0100F0	3800	263		MOV.B	R0L,@CTRLMD
264	0100F2	7F2F7070	264		BSET	#7,@DTCR0B ;DMAC1 start
265	0100F6	58000168	265		BRA	EXIT
266			266			;
267			267			;***** NEXT STOP *****
268			268			;
269	0100FA	7F277270	269	NEXT_S_CW	BCLR	#7,@DTCR0A ;Stop DMAC channel0
270	0100FE	F858	270		MOV.B	#B'01011000,R0L ;Store repeat mode ;data to DTCR0B
271	010100	382F	271		MOV.B	R0L,@DTCR0B
272	010102	790003E8	272		MOV.W	#D'1000,R0 ;Store transfer count ;to ETCR0B
273	010106	6B80FF2C	273		MOV.W	R0,@ETCR0B
274	01010A	7A0000010358	274		MOV.L	#CNSTBL,ER0 ;Store data address to ;MAR0B
275	010110	01006B80FF29	275		MOV.L	ER0,@MAR0B
276	010116	F808	276		MOV.B	##'00,R0L ;Set mode flag
277	010118	3800	277		MOV.B	R0L,@CTRLMD
278	01011A	7F2F7070	278		BSET	#7,@DTCR0B ;DMAC channel1 start
279	01011E	58000140	279		BRA	EXIT
280			280			;
281			281			;***** NEXT SLUJ UP CCW *****
282			282			;
283	010122	F848	283	NEXT_U_CCW	MOV.B	#B'01001000,R0L ;Set MAR0B increment
284	010124	382F	284		MOV.B	R0L,@DTCR0B
285	010126	7900007B	285		MOV.W	#D'120,R0 ;Store Transfer count ;to ETCR0B
286	01012A	6B80FF2C	286		MOV.W	R0,@ETCR0B
287	01012E	7EA47370	287		BTST	#7,@NDRB ;Set NDRB

288	010132	58600010	288	BNE	PTRN1_2CCW	
289	010136	7EA47360	289	BTST	#6,@NDRB	
290	01013A	58600020	290	BNE	PTRN3CCW	
291	01013E	F860	291	MOV.B	#H'60,R0L	
292	010140	38A4	292	MOV.B	R0L,@NDRB	
293	010142	5800001C	293	BRA	NEXTCCW	
294			294	;		
295	010146	7EA47360	295	PTRN1_2CCW BTST	#6,@NDRB	
296	01014A	58600008	296	BNE	PTRN2CCW	
297	01014E	F830	297	MOV.B	#H'30.R0L	
298	010150	38A4	298	MOV.B	R0L,@NDRB	
299	010152	5800000C	299	BRA	NEXTCCW	
300			300	;		
301	010156	F890	301	PTRN2CCW MOV.B	#H'90,R0L	
302	010158	38A4	302	MOV.B	R0L,@NDRB	
303	01015A	58000004	303	BRA	NEXTCCW	
304			304	;		
305	01015E	F8C0	305	PTRN3CCW MOV.B	#H'C0,R0L	
306	010160	38A4	306	MOV.B	R0L,@NDRB	
307	010162	F8B0	307	NEXTCCW MOV.B	#B'10110000,R0L	;MAR0A decrement
308	010164	3827	308	MOV.B	R0L,@DTCR0A	
309	010166	2024	309	MOV.B	@ETCR0AH,R0H	
310	010168	F805	310	MOV.B	#D'5,R0L	
311	01016A	1808	311	SUB.B	R0H,R0L	
312	01016C	3824	312	MOV.B	R0L,@ETCR0AH	
313	01016E	7A000001026B	313	MOV.L	#UPTBL,ER0	;Set data address
314	010174	01006B80FF29	314	MOV.L	ER0,@MAR0B	
315	01017A	F881	315	MOV.B	#H'81,R0L	;Set mode flag
316	01017C	3800	316	MOV.B	R0L,@CTRLMD	
317	01017E	7F2F7070	317	BSET	#7,@DTCR0B	;DMAC channell start
318	010182	7F277070	318	BSET	#7,@DTCR0A	;DTCR0A start
319	010186	580000D8	319	BRA	EXIT	
320			320	;		
321			321	;	***** NEXT CONSTANT CCW *****	
322			322	;		
323	01018A	F858	323	NEXT_C_CCW MOV.B	#B'01011000,R0L	;Store repeat mode ;data to DTCR0B
324	01018C	382F	324	MOV.B	R0L,@DTCR0B	
325	01018E	79000BB8	325	MOV.W	#D'3000,R0	;Store transfer count ;to ETCR0B
326	010192	6B80FF2C	326	MOV.W	R0,@ETCR0B	
327	010196	7A0000010358	327	MOV.L	#CNSTBL,ER0	;Store data address to ;MAR0B
328	01019C	01006B80FF29	328	MOV.L	ER0,@MAR0B	
329	0101A2	F882	329	MOV.B	#H'82,R0L	;Set mode flag
330	0101A4	3800	330	MOV.B	R0L,@CTRLMD	

331	0101A6	7F2F7070	331	BSET	#7,@DTCR0B	;DMAC channel1 start
332	0101AA	580000B4	332	BRA	EXIT	
333			333	;		
334			334	;***** NEXT SLUE DOWN CCW *****		
335			335	;		
336	0101AE	F868	336	NEXT_D_CCW	MOV.B	#B'01101000,R0L ;Set MAR0B decrement
337	0101B0	382F	337		MOV.B	R0L,@DTCR0B
338	0101B2	79000078	338		MOV.W	#D'120,R0 ;Store transfer count ;to ETCR0B
339	0101B6	6B80FF2C	339		MOV.W	R0,@ETCR0B
340	0101BA	7A0000010356	340		MOV.L	#DOWNTEL,ER0 ;Store data address to ;MAR0B
341	0101C0	01006B80FF29	341		MOV.L	ER0,@MAR0B
342	0101C6	F883	342		MOV.B	#H83,R0L ;Set mode flag
343	0101C8	3800	343		MOV.B	R0L,@CTRLMD
344	0101CA	7F2F7070	344	BSET	#7,@DTCR0B	;DMAC channel1 start
345	0101CE	58000090	345	BRA	EXIT	
346			346	;		
347			347	;***** NEXT STOP *****		
348			348	;		
349	0101D2	7F277270	349	NEXT_S_CCW	BCLR	#7,@DTCR0A ;Stop DMAC channel0
350	0101D6	F858	350		MOV.B	#B'01011000,R0L ;Store repeat mode ;data to DTCR0B
351	0101D8	382F	351		MOV.B	R0L,@DTCR0B
352	0101DA	790003E8	352		MOV.W	#D'1000,R0 ;Store transfer count ;to ETCR0B
353	0101DE	6B80FF2C	353		MOV.W	R0,@ETCR0B
354	0101E2	7A0000010358	354		MOV.L	#CNSTBL,ER0 ;Store data address to ;MAR0B
355	0101E8	01006B80FF29	355		MOV.L	ER0,@MAR0B
356	0101EE	F880	356		MOV.B	#H'80,R0L ;Set mode flag
357	0101F0	3800	357		MOV.B	R0L,@CTRLMD
358	0101F2	7F2F7070	358	BSET	#7,@DTCR0B	;DMAC channel1 start
359	0101F6	58000058	359	BRA	EXIT	
360			360	;		
361			361	;***** NEXT SLUE UP CW *****		
362			362	;		
363	0101FA	F848	363	NEXT_U_CCW	MOV.B	#B01001000,R0L ;Set MAR0B increment
364	0101FC	382F	364		MOV.B	R0L,@DTCR0B
365	0101FE	79000078	365		MOV.W	#D'120,R0 ;Store transfer count ;to ETCR0B
366	010202	6B80FF2C	366		MOV.W	R0,@ETCR0B
367	010206	7EA47370	367	BTST	#7,@NDRB	;Set NDRB
368	01020A	58600010	368	BNE	PTRN1_2CW	
369	01020E	7EA47360	369	BTST	#6,@NDRB	
370	010212	58600020	370	BNE	PTRN3CW	

```

371 010216 F890 371 MOV.B #H'90,R0L
372 010218 38A4 372 MOV.B R0L,@NDRB
373 01021A 5800001C 373 BRA NEXTCW
374 374 ;
375 01021E 7EA47360 375 PTRN1_2CW BTST #6,@NDRB
376 010222 58600008 376 BNE PTRN2CW
377 010226 F8C0 377 MOV.B #H'CO,R0L
378 010228 38A4 378 MOV.B R0L,@NDRB
379 01022A 5800000C 379 BRA NEXTCW
380 380 ;
381 01022E F860 381 PTRN2CW MOV.B #H'60,R0L
382 010230 38A4 382 MOV.B R0L,@NDRB
383 010232 58000004 383 BRA NEXTCW
384 384 ;
385 010236 F830 385 PTRN3CW MOV.B #H'30,R0L
386 010238 38A4 386 MOV.B R0L,@NDRB
387 01023A F890 387 NEXTCW MOV.B #B'10010000,R0L ;MAR0 increment
388 01023C 3827 388 MOV.B R0L,@DTCR0A
389 01023E 2024 389 MOV.B @ETCR0AH,R0H
390 010240 F805 390 MOV.B #D'5,R0L
391 010242 1808 391 SUB.B R0H,R0L
392 010244 3824 392 MOV.B R0L,@ETCR0AH
393 010246 7A0000010268 393 MOV.L #UPTBL,ER0 ;Store data address to
;MAR0B
394 01024C 01006B80FF29 394 MOV.L ER0,@MAR0B
395 010252 F801 395 MOV.B #H'01,R0L ;Set mode flag
396 010254 3800 396 MOV.B R0L,@CTRLMD
397 010256 7F2F7070 397 BSET #7,@DTCR0B ;DMAC channel1 start
398 01025A 7F277070 398 BSET #7,@DTCR0A ;DMAC channel0 start
399 01025E 58000000 399 BRA EXIT
400 400 ;
401 010262 01006D70 401 EXIT POP.L ER0
402 010266 5670 402 RTE
403 403 ;
404 404 ;*****
405 405 ;* *
406 406 ;* DATA TABLE (TIMER PERIOD) *
407 407 ;* *
408 408 ;*****
409 409 ;
410 410 ;***** SLUE UP & DOWN DATA *****
411 411 ;
412 00010268 412 UPTBL .EQU $
413 010268 F424B1646EA361D2 413 .DATA.W D'62500,D'45412,D'28323,D'25042,D'21761,D'20058,
414 010270 55014E5A47B34371 414 + D'18355,D'17265,D'16175,D'15401,D'14626,D'14039
010278 3F2F3C29392236D7

```

415	010280	348C32BB30EA2F6E	415	.DATA.W	D'13452,D'12987,D'12522,D'12142,D'11762,D'11444,
416	010288	2DF22CB42B752A66	416	+	D'11125,D'10854,D'10582,D'10347,D'10111,D'9905
	010290	2956286B277F26B1			
417	010298	25E226BB247423D1	417	.DATA.W	D'9698,D'9915,D'9332,D'9169,D'9005,D'8857,
418	0102A0	232D22992205217F	418	+	D'8709,D'8575,D'8441,D'8319,D'8196,D'8084
	0102A8	20F9207F20041F94			
419	0102B0	1F241EBD1E551DF5	419	.DATA.W	D'7972,D'7869,D'7765,D'7669,D'7573,D'7484,
420	0120B8	1D951D3C1CE31C90	420	+	D'7395,D'7312,D'7229,D'7151,D'7073,D'7000
	0102C0	1C3D1BEF1BA11B58			
421	0102C8	1B0F1ACB1A861A46	421	.DATA.W	D'6927,D'6859,D'6790,D'6726,D'6661,D'6600,
422	0102D0	1A0519C8198B1951	422	+	D'6539,D'6481,D'6423,D'6368,D'6313,D'6261
	0102d8	191718e018a91875			
423	0102E0	1841181017DE17AF	423	.DATA.W	D'6209,D'6160,D'6110,D'6063,D'6015,D'5970,
424	0102E8	177F1752172416F9	424	+	D'5924,D'5881,D'5838,D'5797,D'5755,D'5716
	0102F0	16CE16A5167B1654			
425	0102F8	162C160615E015BB	425	.DATA.W	D'5676,D'5638,D'5600,D'5563,D'5526,D'5491,
426	010300	159615731551152E	426	+	D'5456,D'5422,D'5388,D'5356,D'5323,D'5292
	010308	150C14EC14CB14AC			
427	010310	148C146E144F1432	427	.DATA.W	D'5260,D'5230,D'5199,D'5170,D'5140,D'5112,
428	010318	141413F813DC13C1	428	+	D'5084,D'5057,D'5029,D'5002,D'4975,D'4950
	010320	13A513BA136F1356			
429	010328	133C1323130A12F2	429	.DATA.W	D'4924,D'4899,D'4874,D'4850,D'4825,D'4802,
430	010330	12D912C212AA1294	430	+	D'4778,D'4756,D'4733,D'4711,D'4688,D'4667
	010338	127D12671250123B			
431	010340	1225121011FB11E7	431	.DATA.W	D'4645,D'4624,D'4603,D'4583,D'4562,D'4542,
432	010348	110211BE11AA1197	432	+	D'4522,D'4503,D'4483,D'4465,D'4446
	010350	11831171115E			
433			433	;	
434		00010356	434	DOWN1356	.EQU \$
435	010356	115E	435	.DATA.W	D'4446
436			436	;	
437			437	***** CONSTANT *****	
438			438	;	
439		00010358	439	CNSTBL	.EQU \$
440	010358	115E	440	.DATA.W	D'4446
441			441		
442			442	*****	
443			443	;* *	
444			444	;* DATA TABLE (OUTPUT PATTERN) *	
445			445	;* *	
446			446	*****	
447			447	;	
448		0001035A	448	PATTBL	.EQU \$
449	01035A	603090C0	449	.DATA.B	H'60,H'30,H'90,H'C0
450			450	;	
451			451	.END	

*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0

3.5 Five-Phase Stepping Motor Application Example

3.5.1 Specifications

- Control of 5-phase stepping motor using H8/3003 on-chip functions of the ITU, TPC and DMAC (see figure 3.31)
- Stepping motor control by 2-phase drive system
- This sample task repeats the operations “stop -> forward -> stop -> reverse -> stop”
- This sample task does not require software intervention to conduct slue-up and slue-down processing
- A through-current prevention interval is introduced to protect the motor driver

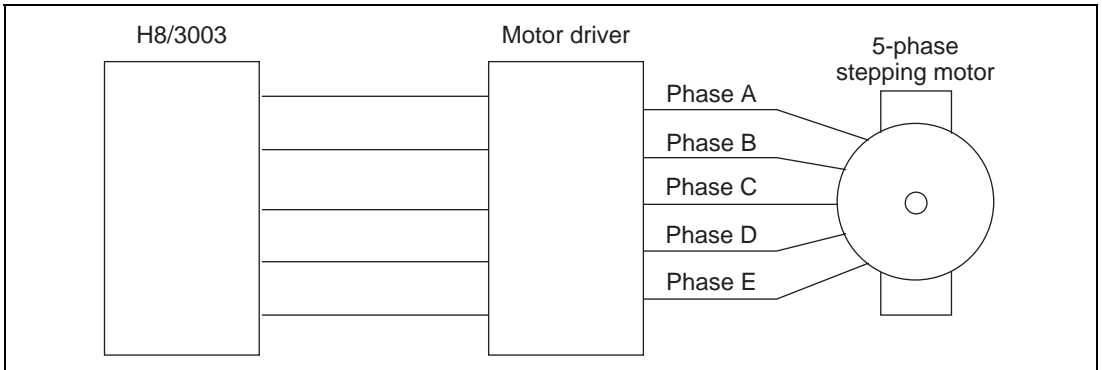


Figure 3.31 5-Phase Stepping Motor Control Circuit Diagram

3.5.2 Method

Example of Stepping Motor Operation: Figure 3.32 shows how 2-phase drive is employed to operate a 5-phase stepping motor. The following is an overview of operations:

- When the pulse is high, the corresponding phase undergoes excitation
- In (1), phase E and phase A undergo simultaneous drive. At this time the rotor is at an intermediate position between phase E and phase A
- In (2), phase A and phase B undergo simultaneous drive. At this time the rotor is between phase A and phase B. The 2-phase drive system then excites the neighboring two phases in sequence to turn the rotor (E-A → A-B → B-C → C-D → D-E)
- In reverse operation, the sequence of excitation is reversed (E-D → D-C → C-B → B-A → A-E)
- Stopping the motor is achieved by excitation for a constant period at the last phase of forward or reverse rotation

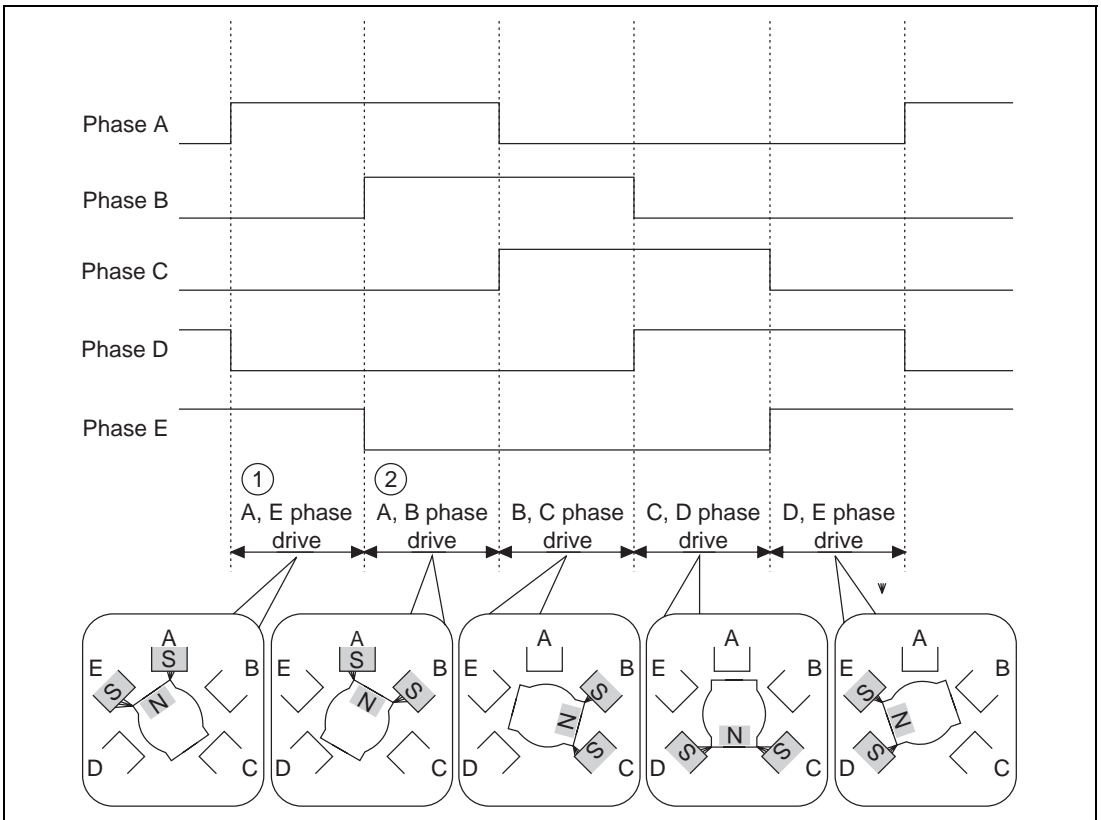


Figure 3.32 Example of Stepping Motor Operation

Non-Overlap Time: As shown in figure 3.33, a through-current prevention period (n) or non-overlap time is inserted during output pattern switching. Because of a turn-off lag created during a drive phase switch, there is a danger of the driver being destroyed. Non-overlap time provides a time delay to counter this.

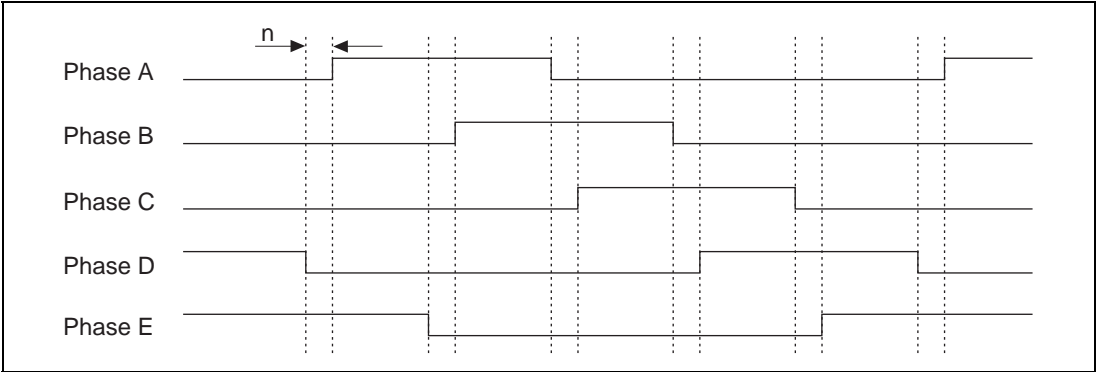


Figure 3.33 Example of Non-Overlap Time Output

Slue-Up and Slue-Down Operation: As shown in figure 3.34, acceleration/deceleration controlled pulses are output. Conducting slue-up and slue-down operation in this way prevents step-out of the motor.

When starting the motor, if there is a sudden output of short pulses, the motor may be unable to take up the load and fail to rotate. Slue-up and slue-down operation is a means of preventing this.

The operating principle is as follows:

1. Gradual shortening of the pulse cycle before outputting the set amount of pulses. (Slue-up operation)
2. Output of set amount of pulses at a constant pulse cycle. (Constant speed operation)
3. Gradual lengthening of the pulse cycle before outputting the set amount of pulses. (Slue-down operation)

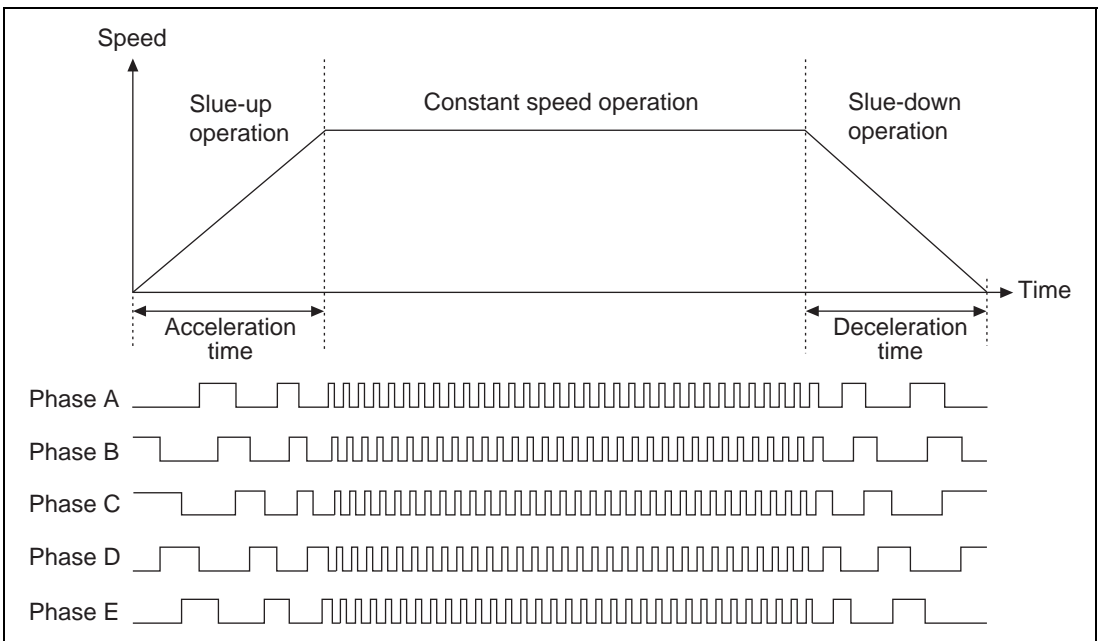


Figure 3.34 Slue-Up and Slue-Down Operation

3.5.3 Description of Functions Used

Figure 3.35 shows the H8/3003 on-chip functions used in this sample task.

Two DMAC channels are used, and a 5-phase output pattern is subjected to timing pattern control (TPC) to give a 5-phase pulse output. Also, the pulse cycle is transferred to the ITU's GRB to control 5-phase pulse output timing control.

Cycle Data Table: Table for altering the step cycle (ITU timer values) set in ROM.

Output Pattern Data Table: Output pattern set in ROM for 5-phase pulse output.

DMAC0A: Starts up by ITU compare/match A. Transfers output pattern from output pattern data table to the TPC's NDR.

DMAC0B: Starts up by ITU compare/match A. Transfers cycle data from cycle data table to the ITU's GRB.

ITU:

- Compare/match A: Starts up DMAC0A, DMAC0B and TPC
- Compare/match B: Clears timer counter and starts up TPC

TPC: Uses the DMAC so data is output without CPU intervention. Although a maximum of 16-bit data can be output, this sample task outputs 5 bit data. Enables output of pulses with non-overlap time.

- Compare/match B: Outputs pulses changing from high to low. Output changing from low to high is held
- Compare/match A: Conducts output of pulse held by compare/match B (L -> H pulse) (delayed by GRA set value)

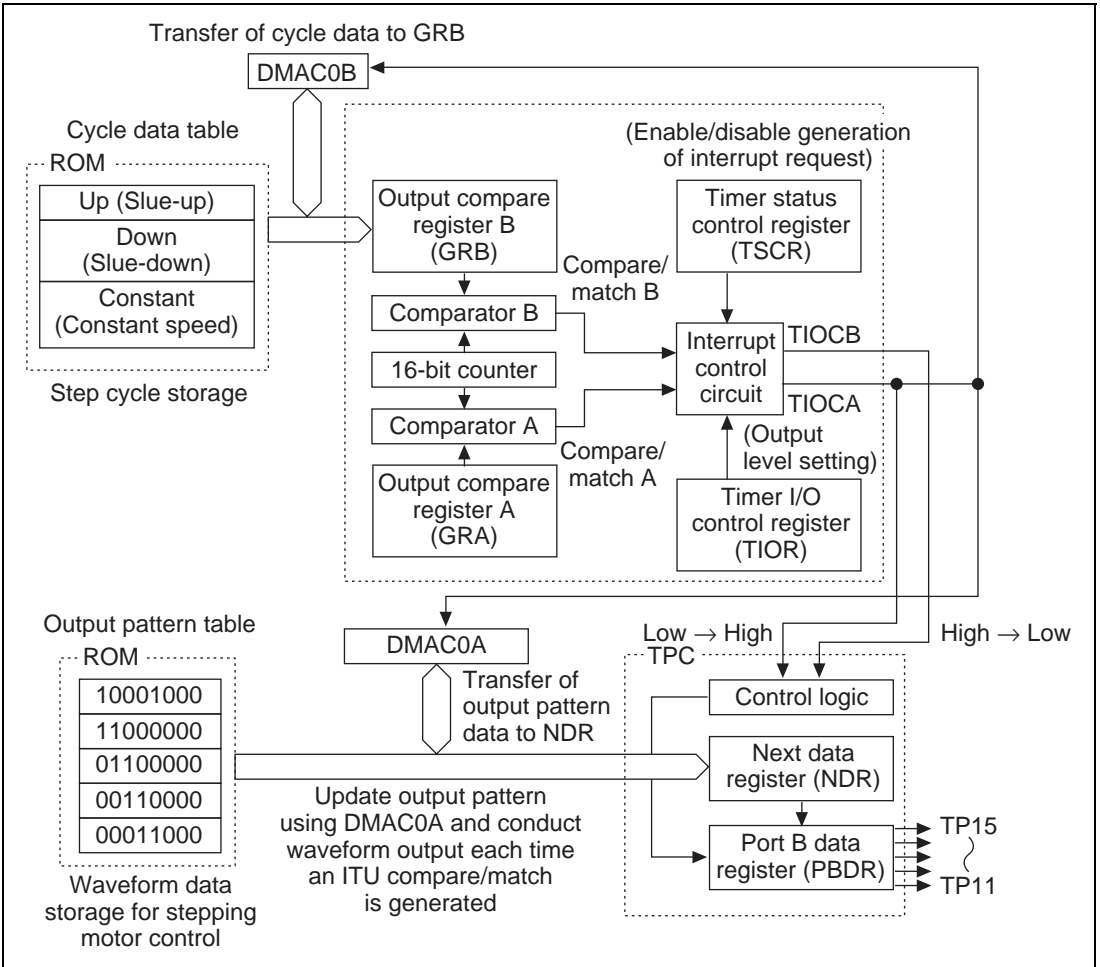


Figure 3.35 Block Diagram of 5-Phase Stepping Motor Control

Table 3.24 shows the function allocation in this sample task. H8/3003 functions are allocated to generate 2-phase drive stepping motor output waveforms.

Table 3.24 H8/3003 Function Allocation

H8/3003 Function	Function	
ITU (ch0)	TCNT0	16-bit counter
	GRA0	Output compare register A
	GRB0	Output compare register B
	TCR0	Selects count clock and sets counter clear factor
	TIOR0	Sets GRA and GRB to output compare registers
	TIER0	Controls interrupt request enable/disable
	TSR0	Compare/match and overflow status register
	TSTR	Enables/disables TCNT counting operation
DMAC0A, B	DTCR0A, B	Controls operation of each DMAC channel
	MAR0A, B	Sets transfer source address
	IORA0A, B	Sets transfer destination address
	ETCR0A, B	Sets number of transfers
TPC	PBDDR	Sets TPC output pins
	PBDR	Stores TPC output B data
	TPMR	Sets TPC output non-overlap mode
	TPCR	Sets TPC output trigger signal
	NDERB	Enables/disables TPC output B
	NDRB	Stores next TPC output B data
	TP15 – TP11	Generates 5-phase stepping motor output waveforms

3.5.4 Description of Operations

5-phase pulse output (with non-overlap time) (figure 3.36).

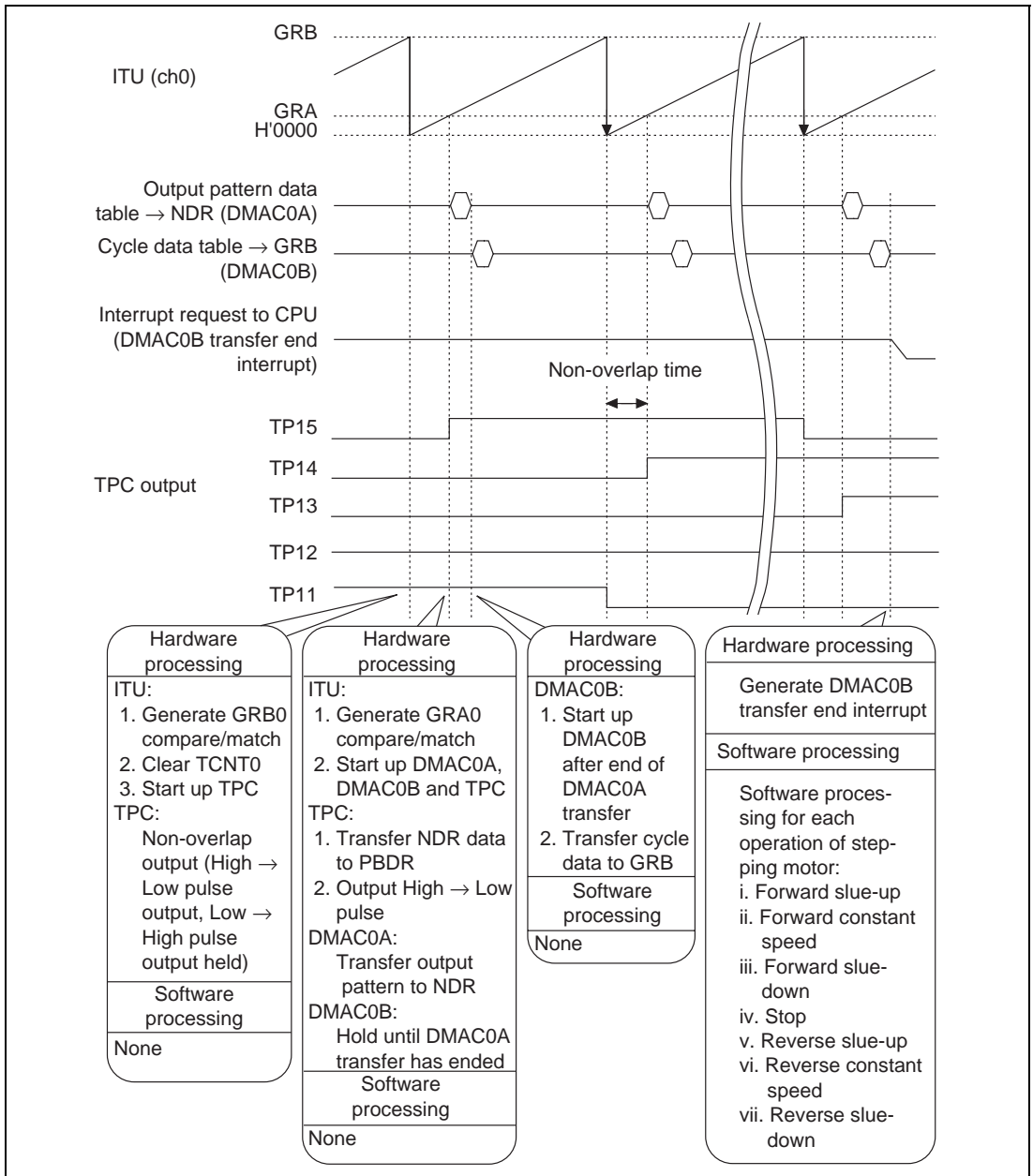


Figure 3.36 Stepping Motor Timing Description

3.5.5 Description of Software

The software for the five-phase stepping motor application example is described below.

Table 3.25 Description of Modules

Module Name	Label Name	Function
Main routine	ST5MIN	ITU, TPC and DMAC0A/DMAC0B initial settings, and forward slue-up operation
Motor control	DENDINT	Motor operations (forward constant speed, stop etc.) control <ul style="list-style-type: none">• Sets DMAC0A/B and conducts forward slue-up operation• Sets DMAC0B and conducts forward constant speed operation• Sets DMAC0A/B and conducts forward slue-down operation• Sets DMAC0A/B and conducts stopping operation• Sets DMAC0A/B and conducts reverse slue-up operation• Sets DMAC0B and conducts reverse constant speed operation• Sets DMAC0A/B and conducts reverse slue-down operation• Sets DMAC0A/B and conducts stopping operation

Table 3.26 Description of Internal Registers

Onchip Function	Register Name	Function
ITU	GRA0	Sets non-overlap time
	GRB0	Sets timer cycle
	TIER0	Enables IMFA interrupts
	TCR0	Sets ITU as follows: <ul style="list-style-type: none"> • Clear counter with GRB compare/match • Count by internal clock ϕ
	TIOR0	Sets GRA and GRB to output compare registers, and prohibits pin output
	TSTR	Enables TCNT0 counting operation
	DMAC0A	DTCR0A
MAR0A		Sets output pattern data table transfer source address
IOAR0A		Sets NDRB address (transfer destination)
ETCR0A		Sets number of transfers
DMAC0B		DTCR0B
	MAR0B	Sets cycle data table transfer source address
	IOAR0B	Sets GRB address (transfer destination)
	ETCR0B	Sets number of transfers

Table 3.26 Description of Internal Registers (cont)

Onchip Function	Register Name	Function
TPC	PBDDR	Enables TP15 – TP11 TPC output
	PBDR	Stores output pattern data
	TPMR	Sets TP15 – TP11 to non-overlap output
	TPCR	Sets TP15 – TP11 output trigger to ITU ch0 compare/match
	NDERB	Enables TP15 – TP11 TPC output
	NDRB	Stores next output pattern data

Description of Arguments: Arguments are not used in this sample task.

Table 3.27 Description of RAM

Label Name	Function	Data Length	Module Name
PDAT_F	Flag indicating operation mode of the 5-phase stepping motor. H'01: indicates forward slue-up operation end H'02: indicates forward constant speed operation end H'04: indicates forward slue-down operation end H'08: indicates forward stop operation end H'10: indicates reverse slue-up operation end H'20: indicates reverse constant speed operation end H'40: indicates reverse slue-down operation end H'80: indicates reverse stop operation end	1 byte	Motor control

Table 3.28 Description of Data Tables

Table Name	Function	Data Length	Data Volume
TBL	Sets pattern for 5-phase pulse output	Byte	5 bytes
UP, DOWN, CONSTANT	Sets data for changing step cycle		2540 bytes

Description of General Registers: Figure 3.37 describes the general registers.

	31	ERn	RnH	RnL	0
0	Used as work register during data setting in all routines				
1					
2					
3					
4					
5					
6					
7					

Figure 3.37 Description of General Registers

3.5.6 Flowcharts

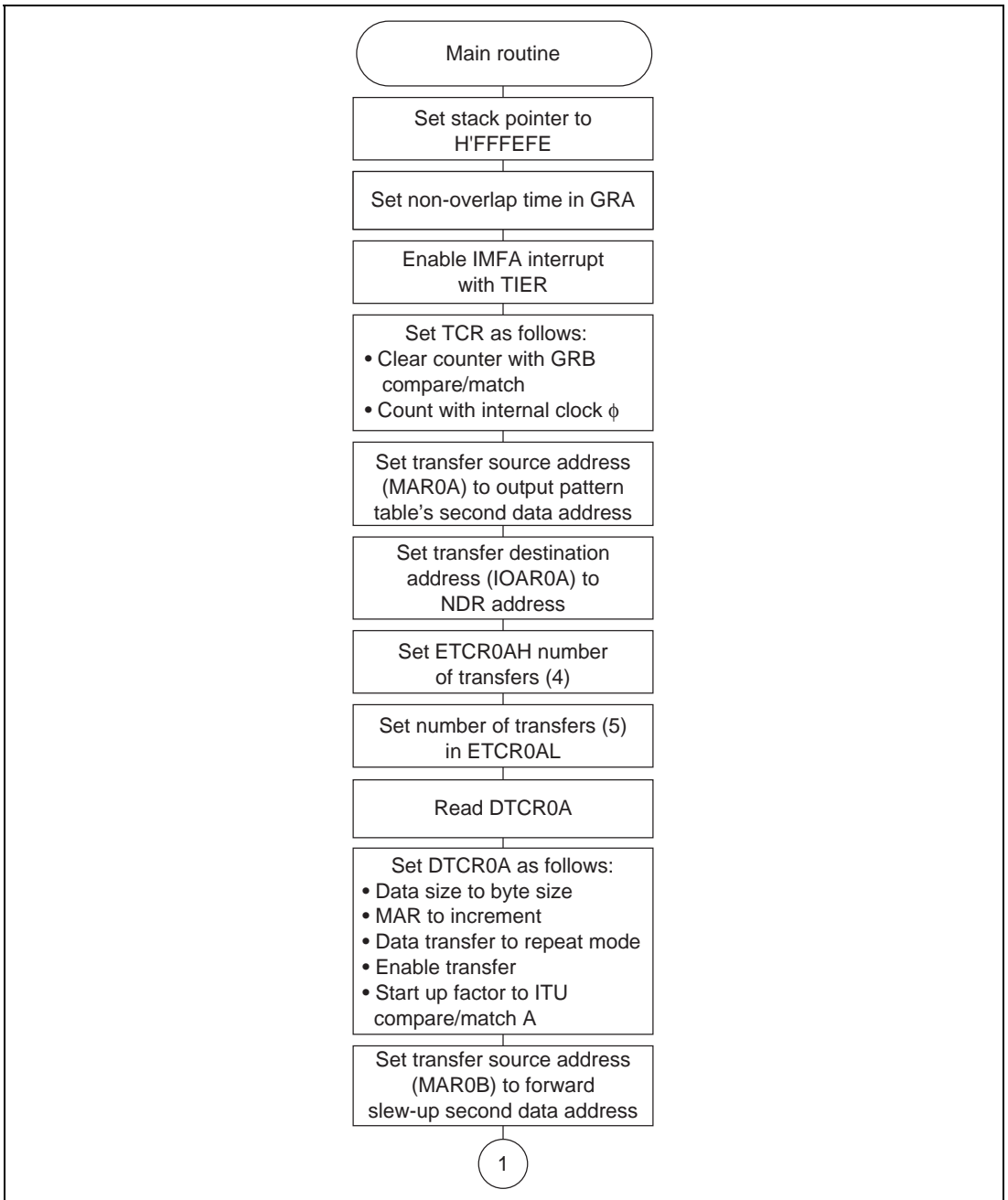


Figure 3.38 Main Routine Flowchart (1)

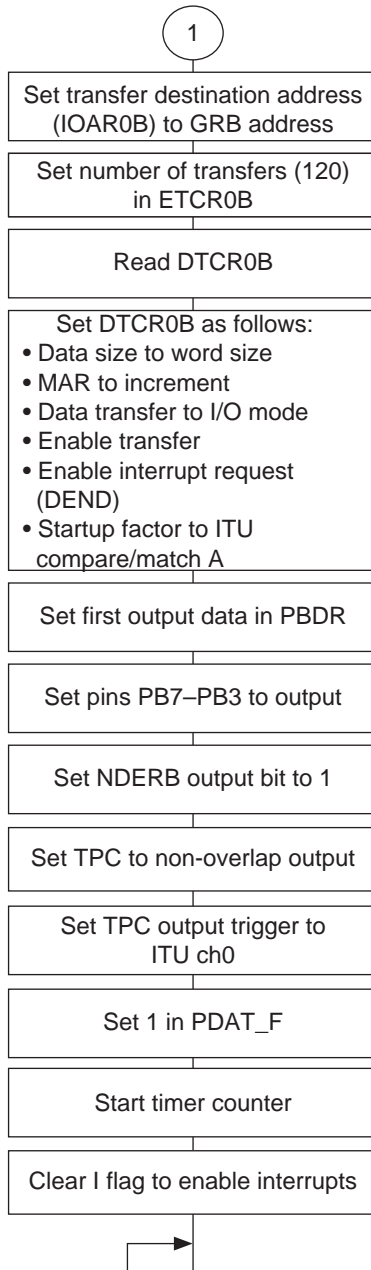


Figure 3.38 Main Routine Flowchart (2)

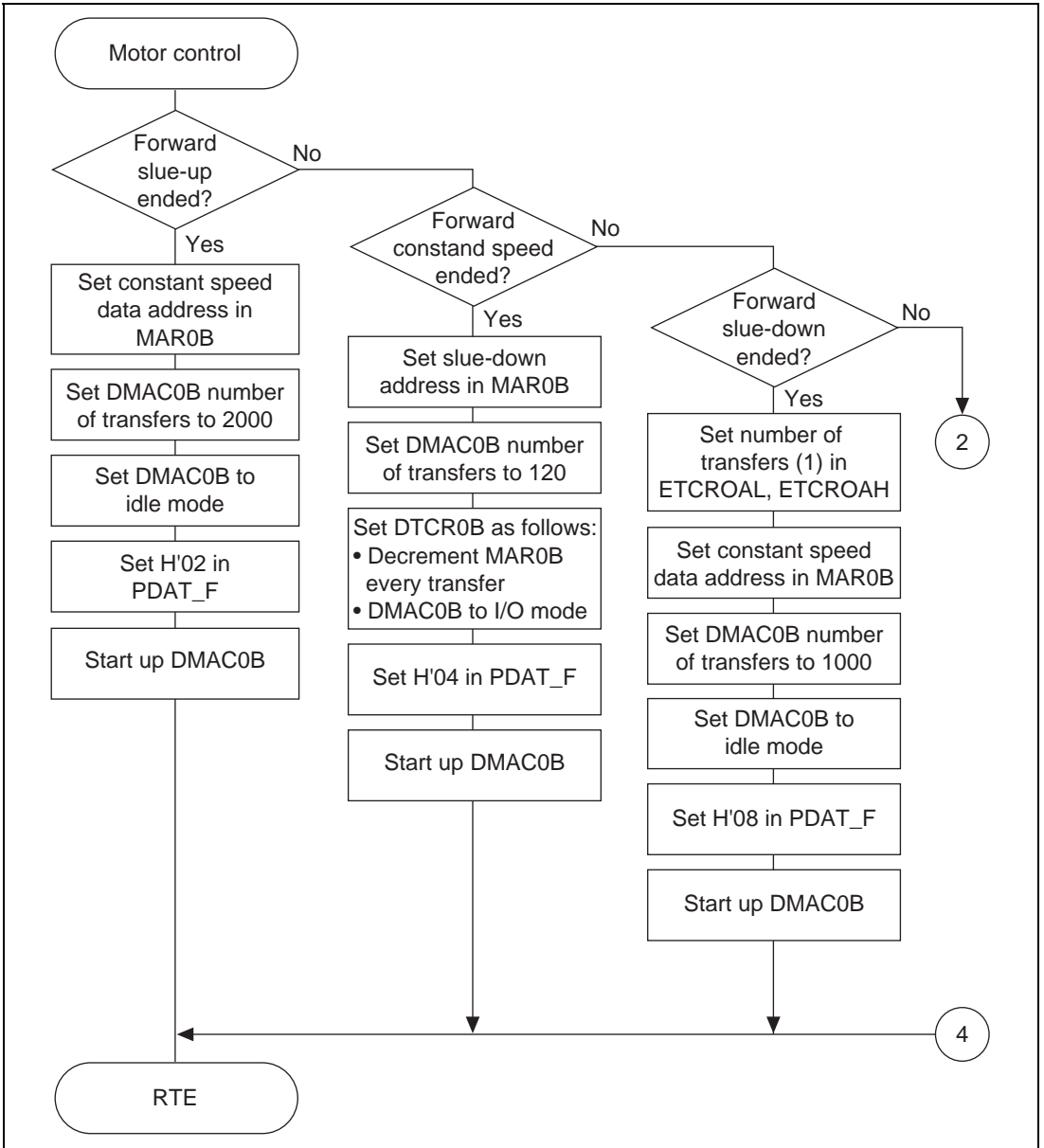


Figure 3.39 Motor Control Flowchart (1)

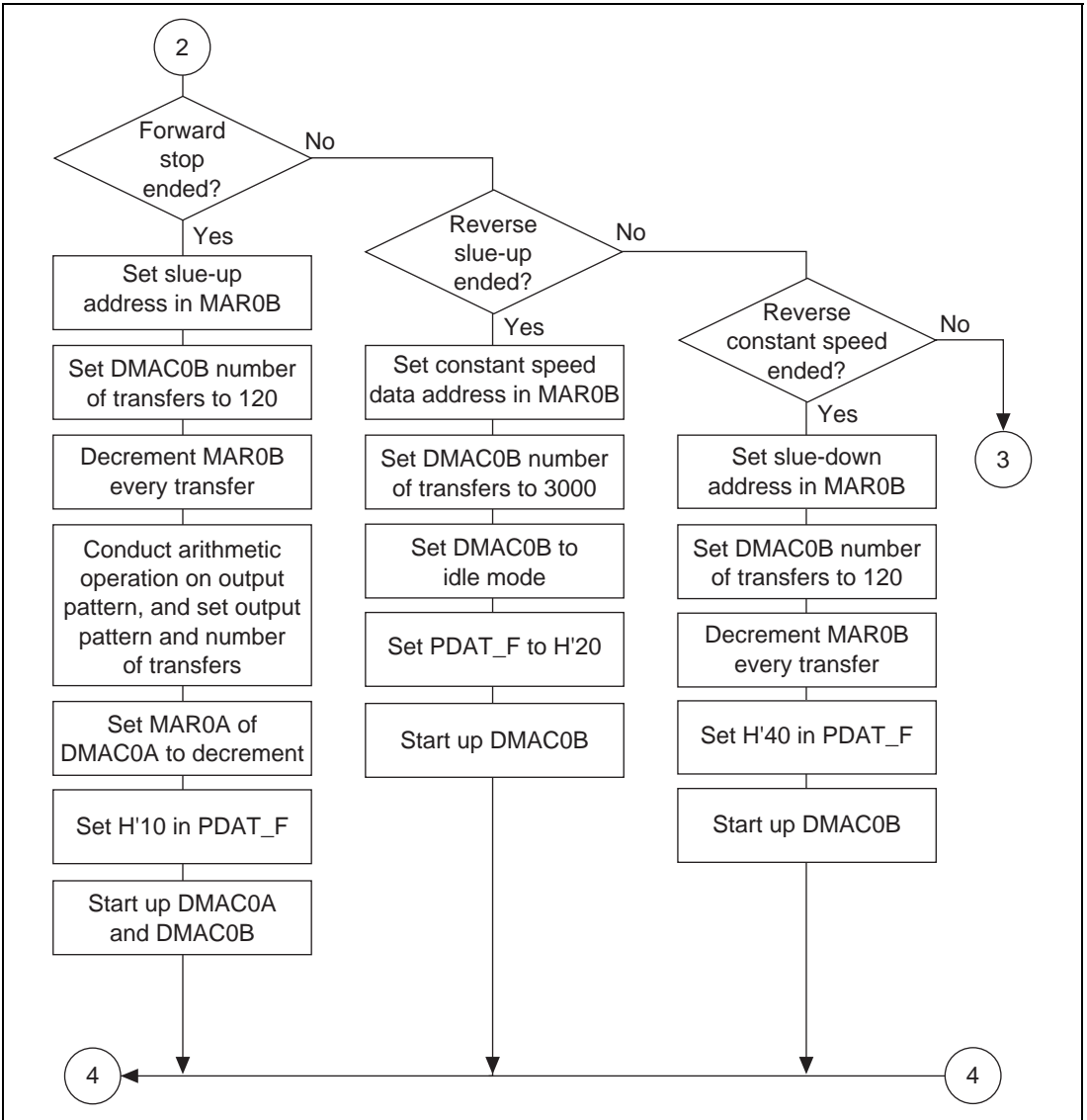


Figure 3.39 Motor Control Flowchart (2)

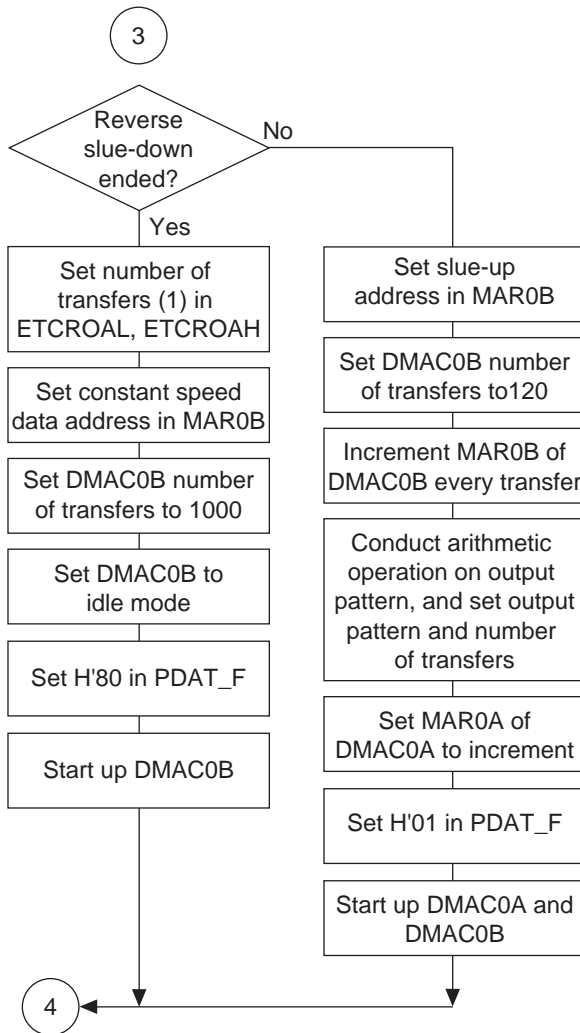


Figure 3.39 Motor Control Flowchart (3)

3.5.7 Program List

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS          *
4          4 ;*
5          5 ;*****
6          6
7          7          .CPU          300HA
8 000000   8          .SECTION          VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000  00010000   10 RES          .DATA.L          STP5MN
11         11 ;
12 00001C   12          .ORG          H'00001C
13 00001C  00010000   13 NMI          .DATA.L          STP5MN
14         14 ;
15 000030   15          .ORG          H'000030
16 000030  00010000   16 IRQ0         .DATA.L          STP5MN
17 000034  00010000   17 IRQ1         .DATA.L          STP5MN
18 000038  00010000   18 IRQ2         .DATA.L          STP5MN
19 00003C  00010000   19 IRQ3         .DATA.L          STP5MN
20 000040  00010000   20 IRQ4         .DATA.L          STP5MN
21 000044  00010000   21 IRQ5         .DATA.L          STP5MN
22 000048  00010000   22 IRQ6         .DATA.L          STP5MN
23 00004C  00010000   23 IRQ7         .DATA.L          STP5MN
24 000050  00010000   24 WOVI0        .DATA.L          STP5MN
25 000054  00010000   25 CMI          .DATA.L          STP5MN
26         26 ;
27 000060   27          .ORG          H'000060
28 000060  00010000   28 IMIA0        .DATA.L          STP5MN
29 000064  00010000   29 IMIB0        .DATA.L          STP5MN
30 000068  00010000   30 OVIO         .DATA.L          STP5MN
31         31 ;
32 000070   32          .ORG          H'000070
33 000070  00010000   33 IMIA1        .DATA.L          STP5MN
34 000074  00010000   34 IMIB1        .DATA.L          STP5MN
35 000078  00010000   35 OV11         .DATA.L          STP5MN
36         36 ;
37 000080   37          .ORG          H'000080
38 000080  00010000   38 IMIA2        .DATA.L          STP5MN
39 000084  00010000   39 IMIB2        .DATA.L          STP5MN
40 000088  00010000   40 OV12         .DATA.L          STP5MN
41         41 ;
42 000090   42          .ORG          H'000090
43 000090  00010000   43 IMIA3        .DATA.L          STP5MN

```



```

44 000094 00010000 44 IMIB3 .DATA.L STP5MN
45 000098 00010000 45 OVI3 .DATA.L STP5MN
46 46 ;
47 0000A0 47 .ORG H'0000A0
48 0000A0 00010000 48 IMIA4 .DATA.L STP5MN
49 0000A4 00010000 49 IMIB4 .DATA.L STP5MN
50 0000A8 00010000 50 OVI4 .DATA.L STP5MN
51 51 ;
52 0000B0 52 .ORG H'0000B0
53 0000B0 00010000 53 DEND0A .DATA.L STP5MN
54 0000B4 00010080 54 DEND0B .DATA.L DENDINT
55 0000B8 00010000 55 DEND1A .DATA.L STP5MN
56 0000BC 00010000 56 DEND1B .DATA.L STP5MN
57 0000C0 00010000 57 DEND2A .DATA.L STP5MN
58 0000C4 00010000 58 DEND2B .DATA.L STP5MN
59 0000C8 00010000 59 DEND3A .DATA.L STP5MN
60 0000CC 00010000 60 DEND3B .DATA.L STP5MN
61 0000D0 00010000 61 ERI0 .DATA.L STP5MN
62 0000D4 00010000 62 RXI0 .DATA.L STP5MN
63 0000D8 00010000 63 TXI0 .DATA.L STP5MN
64 0000DC 00010000 64 TEI0 .DATA.L STP5MN
65 0000E0 00010000 65 ERI1 .DATA.L STP5MN
66 0000E4 00010000 66 RXI1 .DATA.L STP5MN
67 0000E8 00010000 67 TXI1 .DATA.L STP5MN
68 0000EC 00010000 68 TEI1 .DATA.L STP5MN
69 0000F0 00010000 69 ADI .DATA.L STP5MN
70 70 ;
71 71 ;*****
72 72 ;* *
73 73 ;* RAM ALLOCATION *
74 74 ;* *
75 75 ;*****
76 76 ;
77 FFFF00 77 .SECTION RAM,DATA,LOCATE=H'FFFF00
78 78 ;
79 FFFF00 00000001 79 PDAT_F .RES.B 1 ;Mod data
80 80 ;
81 81 ;*****
82 82 ; * *
83 83 ;* SYMBOL DEFINITIONS *
84 84 ;* *
85 85 ;*****
86 86 ;
87 87 ; ***** DMAC0A *****
88 88 ;
89 00FFFF21 89 MAR0AE: .EQU H'FFFF21 ;Memory address register

```

```

90      00FFFF24      90  ETCR00AH:  .EQU      H'FFFF24      ;Transmit cunt register 'H'
91      00FFFF25      91  ETCR0AL:  .EQU      H'FFFF25      ;Transmit cunt register 'L'
92      00FFFF26      92  IOAR0A:  .EQU      H'FFFF26      ;I/O address register
93      00FFFF27      93  DTCR0A:  .EQU      H'FFFF27      ;Data transfer control
                                           ;register
94      00000007      94  DTE0A:  .EQU      7              ;Data transfer enable bit
95      ;
96      ;***** DMACOB *****
97      00FFFF28      97  MAR0BE:  .EQU      H'FFFF28      ;Memory address register
98      00FFFF2C      98  ETCR0BH:  .EQU      H'FFFF2C      ;Transmit cunt register
99      00FFFF2E      99  IOAR0B:  .EQU      H'FFFF2E      ;I/O address register
100     00FFFF2F      100 DTCR0B:  .EQU      H'FFFF2F      ;Data transfer control
                                           ;register
101     00000007      101 DTE0B:  .EQU      7              ;Ddata transfer enable bit
102     ;
103     ;***** ITU *****
104     00FFFF60      104 TSTR:    .EQU      H'FFFF60      ;Timer start registre
105     ;
106     ;***** ITU ch0 *****
107     00FFFF64      107 TCR0:    .EQU      H'FFFF64      ;timer control register
108     00FFFF65      108 TIOR0:  .EQU      H'FFFF65      ;timer I/O control register
109     00FFFF66      109 TIER0:  .EQU      H'FFFF66      ;timer interrupt enable
                                           ;register
110     00FFFF67      110 TSR0:   .EQU      H'FFFF67      ;timer status register
111     00FFFF6A      111 GRA0H:  .EQU      H'FFFF6A      ;General register
112     ;
113     ;***** TPC *****
114     00FFFA0      114 TPMR:   .EQU      H'FFFA0      ;TPC output mod register
115     00FFFA1      115 TPCR:   .EQU      H'FFFA1      ;TPC output control
                                           ;register
116     00FFFA2      116 NDERB:  .EQU      H'FFFA2      ;Next data enable register
                                           ;B
117     00FFFA4      117 NDRB:   .EQU      H'FFFA4      ;Next data register B
118     00FFFD4      118 PBDDR:  .EQU      H'FFFD4      ;Port B data direction
                                           ;register
119     00FFFD6      119 PBDR:   .EQU      H'FFFD6      ;Port B data register
120     ;
121     ;*****
122     ;*
123     ;*      MAIN PROGRAM : STPSMN      *
124     ;*
125     ;*****
126     ;
127     010000      127      .SECTION      PROG.CODE,LOCATE='010000
128     ;
129     00010000      129 STP5MN  .EQU      $

```

```

130 010000 7A0700FFFEFE 130 MOV.L ##'FFFEFE,SP ;Initialize stack pointer
131 131 ;
132 132 ;***** INITIALIZE ITU *****
133 010006 7900000A 133 MOV.W ##'000A,R0
134 01000A 6B00FF6A 134 MOV.W R0,@GRA0H ;Set non overiap time
135 01000E F801 135 MOV.B ##'01,R0L
136 010010 3866 136 MOV.B R0L,@TIER0 ;initialize TIER0(1->IMIEA)
137 010012 F843 137 MOV.B ##'43,R0L
138 010014 3864 138 MOV.B R0L,@TCR0 ;initialize TCR0(1->CCLR1)
139 139 ;
140 140 ;***** INITIALIZE DMAC0A *****
141 010016 7A00000103D2 141 MOV.L #TEL,ER0
142 01001C 0B70 142 INC.L #1,ER0
143 01001E 01006B80FF21 143 MOV.L ER0,@MAR0AE ;SET MAR0A
144 010024 F8A4 144 MOV.B ##'A4,R0L
145 010026 3826 145 MOV.B R0L,@IOAR0A ;SET IOAR0A
146 010028 F804 146 MOV.B ##'04,R0L
147 01002A 3824 147 MOV.B R0L,@ETCR0AH ;STORE FIRST TRANSFER
;COUNT
148 01002C F805 148 MOV.B ##'05,R0L
149 01002E 3825 149 MOV.B R0L,@ETCR0AL ;STORE TRANSFER COUNT
150 010030 2827 150 MOV.B @DTCR0A,R0L ;READ DTCR0A
151 010032 F810 151 MOV.B ##'10,R0L
152 010034 3827 152 MOV.B R0L,@DTCR0A ;INITIALZE DTCR0A
153 010036 7F277070 153 BSET #DTE0A,@DTCR0A ;START DMAC0A
154 154 ;
155 155 ;***** INITIALIZE DMAC0B *****
156 01003A 7A00000102E0 156 MOV.L #UP,ER0
157 010040 0B70 157 INC.L #1,ER0
158 010042 01006B80FF28 158 MOV.L ER0,@MAR0BE ;SET MAR0B
159 010048 F86C 159 MOV.B ##'6C,R0L
160 01004A 382E 160 MOV.B R0L,@IOAR0B ;SET IOAR0B
161 01004C 79000078 161 MOV.W #D'120,R0
162 010050 6B80FF2C 162 MOV.W R0,@ETCR0BH ;STORE TRANSFER COUNT
163 010054 282F 163 MOV.B @DTCR0B,R0L ;READ DTCR0B
164 010056 F848 164 MOV.B ##'48,R0L
165 010058 382F 165 MOV.B R0L,@DTCR0B ;INITIALZE DTCR0B
166 01005A 7F2F7070 166 BSET #DTE0B,@DTCR0B ;START DMAC0B
167 167 ;
168 168 ;***** INITIALIZE TPC *****
169 01005E F888 169 MOV.B ##'88,R0L
170 010060 38D6 170 MOV.B R0L,@PBDR ;SET FIRST OUTPUT DATA
171 010062 F8F8 171 MOV.B ##'F8,R0L
172 010064 38D4 172 MOV.B R0L,@PBDDR
173 010066 38A2 173 MOV.B R0L,@NDERB ;SET OUTPUT TPC
174 010068 F8FC 174 MOV.B ##'FC,R0L

```

```

175 01006A 38A0 175 MOV.B R0L,@TPMR ;SET OUTPUT NON
;OVERLAP

176 01006C F80F 176 MOV.B #H'0F,R0L

177 01006E 38A1 177 MOV.B R0L,@TPCR ;INITIALZE TPCR

178 178 ;

179 010070 F801 179 MOV.B #H'01,R0L

180 010072 3800 180 MOV.B R0L,@PDAT_F ;SET PDAT_F(01)

181 010074 F8E1 181 MOV.B #H'E1,R0L

182 010076 3860 182 MOV.B R0L,@TSTR ;START ITU ch0

183 010078 0700 183 LOC.B #0,CCR ;Enable interrupt

184 184 ;

185 01007A 0000 185 LOOP: NOP

186 01007C 0000 186 NOP

187 01007E 40FA 187 BRA LOOP

188 188 ;

189 189 ;*****
190 190 ;* *
191 191 ;* DENDINT : (DMAC0B END INTERRUPT PROGRAM) *
192 192 ;* *
193 193 ;*****
194 194 ;* *
195 195 ;* ENTRY : NOTHING *
196 196 ;* RETURN : NOTHING *
197 197 ;* *
198 198 ;*****
199 199 ;

200 010080 01006DF0 200 DENDINT: PUSH,L ERO
201 010084 2800 201 MOV.B @PDAT_F,R0L
202 010086 A880 202 CMP.B #H'80,R0L
203 010088 587001B6 203 BEQ DENDINT13
204 204 ;

205 01008C A801 205 CMP.B #H'01,R0L
206 01008E 58700024 206 BEQ DENDINT01
207 207 ;

208 010092 A802 208 CMP.B #H'02,R0L
209 010094 58700044 209 BEQ DENDINT02
210 210 ;

211 010098 A804 211 CMP.B #H'04,R0L
212 01009A 58700064 212 BEQ DENDINT03
213 213 ;

214 01009E A808 214 CMP.B #H'08,R0L
215 0100A0 5870008A 215 BEQ DENDINT04
216 216 ;

217 0100A4 A810 217 CMP.B #H'10,R0L
218 0100A6 58700120 218 BEQ DENDINT10
219 219 ;

```

220	0100AA	A820	220	CMP.B	##'20,R0L	
221	0100AC	58700140	221	BEQ	DENDINT11	
222			222	;		
223	0100B0	A840	223	CMP.B	##'40,R0L	
224	0100B2	58700160	224	BEQ	DENDINT12	
225			225	;		
226			226	;	***** NEXT CONSTANT CW *****	
227			227	;		
228	0100B6	7A00000103D0	288	DENDINT01: MOV.L	#CONSTANT,ER0	
229	0100BC	01006B80FF28	229	MOV.L	ER0,@MAR0BE	;SET DATA ADDRESS
230	0100C2	790007D0	230	MOV.W	#D'2000,R0	
231	0100C6	6B80FF2C	231	MOV.W	R0,@ETCR0BH	;STORE TRANSFER COUNT
232	0100CA	282F	232	MOV.B	@DTCR0B,R0L	;READ DTCR0B
233	0100CC	F818	233	MOV.B	##'18,R0L	
234	0100CE	382F	234	MOV.B	R0L,@DTCR0B	;DMAC08 IDOL MODE
235	0100D0	F802	235	MOV.B	##'02,R0L	
236	0100D2	3800	236	MOV.B	R0L,@PDAT_F	;SET PDAT_F(02)
237	0100D4	7F2F7070	237	BSET	#DTE0B,@DTCR0B	;START DMAC0B
238	0100D8	580001FE	238	BRA	DENDINT99	
239			239	;		
240			240	;	***** NEXT SLUE DOWN CW *****	
241			241	;		
242	0100DC	7A00000103CE	242	DENDINT02: MOV.L	#DOWN,ER0	
243	0100E2	01006B80FF28	243	MOV.L	ER0,@MAR0BE	;Set data address
244	0100E8	79000078	244	MOV.W	#D'120,R0	
245	0100EC	6B80FF2C	245	MOV.W	R0,@ETCR0BH	;Store transfer count
246	0100F0	282F	246	MOV.B	@DTCR0B,R0L	;READ DTCR0B
247	0100F2	F868	247	MOV.B	##'68,R0L	
248	0100F4	382F	248	MOV.B	R0L,@DTCR0B	;Set DTCR0B ;(DTID->1,DTIE->1)
249	0100F6	F804	249	MOV.B	##'04,R0L	
250	0100F8	3800	250	MOV.B	R0L,@PDAT_F	;Set PDAT_F(04)
251	0100FA	7F2F7070	251	BSET	#DTE0B,@DTCR0B	
252			252			
253	0100FE	580001D8	253	BRA	DENDINT99	
254			254	;		
255			255	;	***** NEXT STOP *****	
256			256	;		
257	010102	F801	257	DENDINT03: MOV.B	##'01,R0L	
258	010104	3825	258	MOV.B	R0L,@ETCR0AL	;STORE FIRST TRANSFER COUNT
259	010105	3824	259	MOV.B	R0L,@ETCR0AH	;STORE TRANSFER COUNT
260			260	;		;Stop DMAC0A
261	010108	7A00000103D0	261	MOV.L	#CONSTANT,ER0	
262	01010E	01006B80FF28	262	MOV.L	ER0,@MAR0BE	;Set data address
263	010114	790003E8	263	MOV.W	#D'1000,R0	
264	010118	6B80FF2C	264	MOV.W	R0,@ETCR0BH	;Store transfer count

265	01011C	282F	265	MOV.B	@DTCR0B,R0L	;Read DTCR0B
266	01011E	F818	266	MOV.B	#H'18,R0L	
267	010120	382F	267	MOV.B	R0L,@DTCR0B	;Set DTCR0B
						;(RPE->1,DTIE->1)
268	010122	F808	268	MOV.B	#H'08,R0L	
269	010124	3800	269	MOV.B	R0L,@PDAT_F	;Set PDAT_F(08)
270	010126	7F2F7070	270	BSET	#DTE0B,@DTCR0B	;Start DMAC0B
271	01012A	580001AC	271	BRA	DENDINT99	
272			272	;		
273			273	;	***** NEXT SLUE UP CCW *****	
274			274	;		
275	01012E	7A00000102E0	275	DENDINT04: MOV.L	#UP,ER0	
276	010134	01006B80FF28	276	MOV.L	ER0,@MAR0BE	;Set data address
277	01013A	79000078	277	MOV.W	#D'120,R0	
278	01013E	6B80FF2C	278	MOV.W	R0,@ETCR0BH	;STORE TRANSFER COUNT
279	010142	282F	279	MOV.B	@DTCR0B,R0L	;Read DTCR0B
280	010144	F848	280	MOV.B	#H'48,R0L	
281	010146	382F	281	MOV.B	R0L,@DTCR0B	;Set DTCR0B(DTIE->1)
282			282	;		
283	010148	7EA47370	283	BTST	#7,@NDRB	;Set NDRB
284	01014C	58600018	284	RNE	DENDINT05	
285	010150	7EA47360	285	BTST	#6,@NDRB	
286	010154	58700028	286	BEQ	DENDINT06	
287	010158	F860	287	MOV.B	#H'60,R0L	
288	01015A	38A4	288	MOV.B	R0L,@NDRB	
289	01015C	F803	289	MOV.B	#H'03,R0L	
290	01015E	3824	290	MOV.B	R0L,@ETCR0AH	
291	010160	F805	291	MOV.B	#H05,R0L	
292	010162	3825	292	MOV.B	R0L,@ETCR0AL	
293	010164	5800004C	293	BRA	DENDINT09	
294			294	;		
295	010168	7EA47350	295	DENDINT05: BTST	#6,@NDRB	
296	01016C	58700028	296	BEQ	DENDINT07	
297	010170	F8C0	297	MOV.B	#H'C0,R0L	
298	010172	38A4	298	MOV.B	R0L,@NDRB	
299	010174	F802	299	MOV.B	#H'02R0L	
300	010176	3824	300	MOV.B	R0L,@ETCR0AH	
301	010178	F805	301	MOV.B	#H'05,R0L	
302	01017A	3825	302	MOV.B	R0L,@ETCR0AL	
303	01017C	58000034	303	BRA	DENDINT09	
304			304	;		
305	010180	7EA47350	305	DENDINT06: BTST	#5,@NDRB	
306	010184	58700020	306	BEQ	DENDINT08	
307	010188	F830	307	MOV.B	#H'30,R0L	
308	01018A	38A4	308	MOV.B	R0L,@NDRB	
309	01018C	F804	309	MOV.B	#H'04,R0L	

310	01018E	3824	310	MOV.B	R0L,@ETCR0AH	
311	010190	F805	311	MOV.B	##'05,R0L	
312	010192	3825	312	MOV.B	R0L,@ETCR0AL	
313	010194	5800001C	313	BRA	DENDINT09	
314			314	;		
315	010198	F888	315	DENDINT07: MOV.B	##'88,R0L	
316	01019A	38A4	316	MOV.B	R0L,@NDRB	
317	01019C	F801	317	MOV.B	#01,R0L	
318	01019E	3824	318	MOV.B	R0L,@ETCR0AH	
319	0101A0	F805	319	MOV.B	##'05,R0L	
320	0101A2	3825	320	MOV.B	R0L,@ETCR0AL	
321	0101A4	5800000C	321	BRA	DENDINT09	
322			322	;		
323	0101A8	F818	323	DENDINT08: MOV.B	##'18,R0L	
324	0101AA	38A4	324	MOV.B	R0L,@NDRB	
325	0101AC	F805	325	MOV.B	#05,R0L	
326	0101AE	3824	326	MOV.B	R0L,@ETCR0AH	
327	0101B0	F805	327	MOV.B	##'05,R0L	
328	0101B2	3825	328	MOV.B	R0L,@ETCR0AL	
329			329	;		
330	0101B4	2827	330	DENDINT09: MOV.B	@DTCR0A,R0L	;Read DTCR0A
331	0101B6	F830	331	MOV.B	##'30,R0L	
332	0101B8	3827	332	MOV.B	R0L,@DTCR0A	;Set DTCR0A ;(DTID->1,RPE->1)
333			333	;		;Start DMAC0A
334			334	;		
335	0101BA	F810	335	MOV.B	##'10,R0L	
336	0101BC	3800	336	MOV.B	R0L,@PDAT_F	;Set PDAT_F(10)
337			337	;		
338	0101BE	7F2F7070	338	BSET	#DTE0B,@DTCR0B	;Start DMAC0B
339	0101C2	7F277070	339	BSET	#DTE0A,@DTCR0A	;Start DMAC0A
340			340	;		
341	0101C6	58000110	341	BRA	DENDINT99	
342			342	;		
343			343	;	***** NEXT CONSTANT CCW *****	
344			344	;		
345	0101CA	7A00000103D0	345	DENDINT10: MOV.L	#CONSTANT,ER0	
346	0101D0	01006B80FF28	346	MOV.L	ER0,@MAR0BE	;SET DATA ADDRESS
347	0101D6	79000BB8	347	MOV.W	#D'3000,R0	
348	0101DA	6B80FF2C	348	MOV.W	R0,@ETCR0BH	;STORE TRANSFER COUNT
349	0101DE	282F	349	MOV.B	@DTCR0B,R0L	;READ DTCR0B
350	0101E0	F818	350	MOV.B	##'18,R0L	
351	0101E2	382F	351	MOV.B	R0L,@DTCR0B	;DMAC0B IDOL MODE
352	0101E4	F820	352	MOV.B	##'20,R0L	
353	0101E6	3800	353	MOV.B	R0L,@PDAT_F	;SET PTAT_F(20)
354	0101E8	7F2F7070	354	BSET	#DTE0B,@DTCR0B	;START DMAC0B

```

355 0101EC 580000EA 355 BRA DENDINT99
356 356 ;
357 357 ; ***** NEXT SLUE DOWN CCW *****
358 358 ;
359 0101F0 7A00000103CE 359 DENDINT11: MOV.L #DOWN,ERO
360 0101F6 01006B80FF28 360 MOV.L ER0,@MAR0BE ;Set data address
361 0101FC 79000078 361 MOV.W #D'120,R0
362 010200 6B80FF2C 362 MOV.W R0,@ETCR0BH ;Store transfer count
363 010204 282F 363 MOV.B @DTCR0B,R0L ;READ DTCR0B
364 010206 F868 364 MOV.B #H'68,R0L
365 010208 382F 365 MOV.B R0L,@DTCR0B ;Set DTCR0B
; (DTID->1,DTIE->1)
366 01029A F840 366 MOV.B #H'40,R0L
367 01020C 3800 367 MOV.B R0L,@PDAT_F ;Set PTAT_F(40)
368 01020E 7F2F7070 368 BSET #DTE0B,@DTCR0B ;Start DMAC0B
369 010212 580000C4 369 BRA DENDINT99
370 370 ;
371 371 ; ***** NEXT STOP *****
372 372 ;
373 010216 F801 373 DENDINT12: MOV.B #H'01,R0L
374 010218 3825 374 MOV.B R0L,@ETCR0AL ;STORE FIRST TRANSFER
;COUNT
375 01021A 3824 375 MOV.B R0L,@ETCR0AH ;STORE TRANSFER COUNT
376 376 ; ;DMAC0A Stop
377 01021C 7A00000103D0 377 MOV.L #CONSTANT,ERO
378 010222 01006B80FF28 378 MOV.L ER0,@MAR0BE ;Set data address
379 010228 790003E8 379 MOV.W #D'1000,R0
380 01022C 6B80FF2C 380 MOV.W R0,@ETCR0BH ;Store transfer count
381 010230 282F 381 MOV.B @DTCR0B,R0L ;Read DTCR0B
382 010232 F818 382 MOV.B #H'18,R0L
383 010234 382F 383 MOV.B R0L,@DTCR0B ;Set DTCR0B
; (RPE->1,DTIE->1)
384 010236 F880 384 MOV.B #H'80,R0L
385 010238 3800 385 MOV.B R0L,@PDAT_F ;Set PDAT_F(80)
386 01023A 7F2F7070 386 BSET #DTE0B,@DTCR0B ;Start DMAC0B
387 01023E 58000098 387 BRA DENDINT99
388 388 ;
389 389 ; ***** NEXT SLUE UP CW *****
390 390 ;
391 010242 7A00000102E0 391 DENDINT13: MOV.L #UP,ERO
392 010248 01006B80FF28 392 MOV.L ER0,@MAR0BE ;Set data address
393 393 ;
394 01024E 79000078 394 MOV.W #D'120,R0
395 010252 6B80FF2C 395 MOV.W R0,@ETCR0BH ;Store transfer count
396 396 ;
397 010256 282F 397 MOV.B @DTCR0B,R0L ;Read DTCR0B

```


398	010258	F848	398	MOV.B	##'48,R0L	
399	01025A	382F	399	MOV.B	R0L,@DTCR0B	;Set DTCR0B(DTIE->1)
400			400			
401	01025C	7EA47370	401	BTST	#7,@NDRB	;Set NDRB
402	010260	58600018	402	BNE	DENDINT14	
403	010264	7EA47360	403	BTST	#6,@NDRB	
404	010268	58700028	404	BEQ	DENDINT15	
405	01026C	F860	405	MOV.B	##'60,R0L	
406	01026E	38A4	406	MOV.B	R0L,@NDRB	
407	010270	F803	407	MOV.B	##'03,R0L	
408	010272	3824	408	MOV.B	R0L,@ETCR0AH	
409	010274	F805	409	MOV.B	##'05,R0L	
410	010276	3825	410	MOV.B	R0L,@ETCR0AL	
411	010278	5800004C	411	BRA	DENDINT18	
412			412			
413	01027C	7EA47360	413	DENDINT14: BTST	#6,@NDRB	
414	010280	58700028	414	BEQ	DENDINT16	
415	010284	F8C0	415	MOV.B	##'C0,R0L	
416	010286	38A4	416	MOV.B	R0L,@NDRB	
417	010288	F804	417	MOV.B	##'04,R0L	
418	01028A	3824	418	MOV.B	R0L,@ETCR0AH	
419	01028C	F805	419	MOV.B	##'05,R0L	
420	01028E	3825	420	MOV.B	R0L,@ETCR0AL	
421	010290	58000034	421	BRA	DENDINT18	
422			422			
423	010294	7EA47350	423	DENDINT15: BTST	#5,@NDRB	
424	010298	58700020	424	BEQ	DENDINT17	
425	01029C	F830	425	MOV.B	##'30,R0L	
426	01029E	38A4	426	MOV.B	R0L,@NDRB	
427	0102A0	F802	427	MOV.B	##'02,R0L	
428	0102A2	3824	428	MOV.B	R0L,@ETCR0AH	
429	0102A4	F805	429	MOV.B	##'05,R0L	
430	0102A6	3825	430	MOV.B	R0L,@ETCR0AL	
431	0102A8	5800001C	431	BRA	DENDINT18	
432			432			
433	0102AC	F888	433	DENDINT16: MOV.B	##'88,R0L	
434	0102AE	38A4	434	MOV.B	R0L,@NDRB	
435	0102B0	F805	435	MOV.B	##'05,R0L	
436	0102B2	3824	436	MOV.B	R0L,@ETCR0AH	
437	0102B4	F805	437	MOV.B	##'05,R0L	
438	0102B6	3825	438	MOV.B	R0L,@ETCR0AL	
439	0102B8	5800000C	439	BRA	DENDINT18	
440			440			
441	0102BC	F818	441	DENDINT17: MOV.B	##'18,R0L	
442	0102BE	38A4	442	MOV.B	R0L,@NDRB	
443	0102C0	F801	443	MOV.B	#01,R0L	

444	0102C2	3824	444	MOV.B	R0L,@ETCR0AH	
445	0102C4	F805	445	MOV.B	#H'05,R0L	
446	0102C6	3825	446	MOV.B	R0L,@ETCR0AL	
447			447			
448			448	;		
449	0102C8	2827	449	DENDINT18: MOV.B	@DTCR0A,R0L	;READ DTCR0A
450	0102CA	F810	450	MOV.B	#H'10,R0L	
451	0102CC	3827	451	MOV.B	R0L,@DTCR0A	;Set DTCR0A(RPE->1)
452			452	;		;Start DMAC0A
453			453	;		
454	0102CE	F801	454	MOV.B	#H'01,R0L	
455	0102D0	3800	455	MOV.B	R0L,@PDAT_F	;Set PDAT_F(01)
456			456	;		
457	0102D2	7F2F7070	457	BSET	#DTE0B,@DTCR0B	;Start DMAC0B
458	0102D6	7F277070	458	BSET	#DTE0A,@DTCR0A	;Start DMAC0A
459			459	;		
460	0102DA	01006D70	460	DENDINT99: POP,L	ERO	
461	0102DE	5670	461	RTE		
462			462	;		
463			463	;	*****	
464			464	;	*	*
465			465	;	DATA TABLE (TIMER PERIOD)	*
466			466	;	*	*
467			467	;	*****	
468			468	;		
469	0102E0	F424B1646EA361D2	469	UP	.DATA.W	D'62500,D'45412,D'28323,D'25042,D'21761,D'20058,
470	0102E8	55014E5A47B34371	470	+		D'18355,D'17265,D'16175,D'15401,D'14626,D'14039
	0102F0	3F2F3C29392236D7				
471	0102F8	348C32BB30EA2F6E	471		.DATA.W	D'13452,D'12987,D'12522,D'12142,D'11762,D'11444,
472	010300	2DF22CB42B752A66	472	+		D'11125,D'10854,D'10582,D'10347,D'10111,D'9905
	010308	2956286B277F26B1				
473	010310	25E226BB247423D1	473		.DATA.W	D'9698,D'9915,D'9332,D'9169,D'9005,D'8857,
474	010318	232D22992205217F	474	+		D'8709,D'8575,D'8441,D'8319,D'8196,D'8084
	010320	20F9207F20041F94				
475	010328	1F241EBD1E551DF5	475		.DATA.W	D'7972,D'7869,D'7765,D'7669,D'7573,D'7484,
476	012330	1D951D3C1CE31C90	476	+		D'7395,D'7312,D'7229,D'7151,D'7073,D'7000
	010338	1C3D1BEF1BA11B58				
477	010340	1B0F1ACB1A861A46	477		.DATA.W	D'6927,D'6859,D'6790,D'6726,D'6661,D'6600,
478	010348	1A0519C8198B1951	478	+		D'6539,D'6481,D'6423,D'6368,D'6313,D'6261
	010350	191718E018A91875				
479	010358	1841181017DE17AF	479		.DATA.W	D'6209,D'6160,D'6110,D'6063,D'6015,D'5970,
480	010360	177F1752172416F9	480	+		D'5924,D'5881,D'5838,D'5797,D'5755,D'5716
	010368	16CE16A5167B1654				
481	010370	162C160615E015BB	481		.DATA.W	D'5676,D'5638,D'5600,D'5563,D'5526,D'5491,
482	010378	159615731550152E	482	+		D'5456,D'5422,D'5388,D'5356,D'5323,D'5292
	010380	150C14EC14CB14AC				

```

483 010388 148C146E144F1432 483 .DATA.W D'5260,D'5230,D'5199,D'5170,D'5140,D'5112,
484 010390 141413F813DC13C1 484 + D'5084,D'5057,D'5029,D'5002,D'4975,D'4950
010398 13A5138A136F1356
485 0103A0 133C1323130A12F2 485 .DATA.W D'4924,D'4899,D'4874,D'4850,D'4825,D'4802,
486 0103A8 12D912C212AA1294 486 + D'4778,D'4756,D'4733,D'4711,D'4688,D'4667
0103B0 127D12671250123B
487 0103B8 1225121011FB11E7 487 .DATA.W D'4645,D'4624,D'4603,D'4583,D'4562,D'4542,
488 0103C0 11D211BE11AA1197 488 + D'4522,D'4503,D'4483,D'4465,D'4446
0103CB 11831171115E
489 0103CE 115E 489 DOWN .DATA.W D'4446
490 490 ;*
491 491 ;***** CONSTANT *****
492 492 ;*
493 0103D0 115E 493 CONSTANT .DATA.W D'4446
494 494 ;*
495 495 ;
496 496 ;*****
497 497 ;* *
498 498 ;* DATA TABLE (OUTPUT PATTERN) *
499 499 ;* *
500 500 ;*****
501 501 ;
502 0103D2 603090C0 502 TBL .DATA.B H'60,H'30,H'90,H'C0
503 503 ;
504 504 .END
****TOTAL ERRORS 0
****TOTAL WARNINGS 0

```

**Renesas 16-bit Single-chip Microcomputer
Application Note
H8/300H Series On-Chip Supporting Modules**

Publication Date: 1st Edition, September, 1994
Rev.3.00, August 23, 2004

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



RENESAS SALES OFFICES

<http://www.renesas.com>

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500 Fax: <1> (408) 382-7501

Renesas Technology Europe Limited.

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, United Kingdom
Tel: <44> (1628) 585 100, Fax: <44> (1628) 585 900

Renesas Technology Europe GmbH

Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

Renesas Technology Hong Kong Ltd.

7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.

FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.

26/F., Ruijin Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.

1, Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001



H8/300H Series On-Chip Supporting Modules Application Note



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ05B0465-0300H