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.

H8/300L SLP Series

Stepper Motor Using Two-Phase Excitation

Introduction

The H8/38024 offers various built-in functions. Of these, P63 to P60, the timer A interval function, and the timer F output compare function control a two-phase stepper motor using two phase excitation.

Target Device

H8/38024

Contents

1. Specifications

- The H8/38024 offers various built-in functions. Of these, P63 to P60, timer A interval function and the timer F output compare match function are used to control a two-phase stepper motor.
- The stepper motor is controlled using two-phase excitation, and subjected to the repeated operations for rotating the stepper motor forward, stopping it, rotating the stepper motor in the reverse direction, and then again stopping it.
- The task realizes slew-up and slew-down processing by using software.
- Figure 1 shows the connection diagram for controlling a two-phase stepper motor.

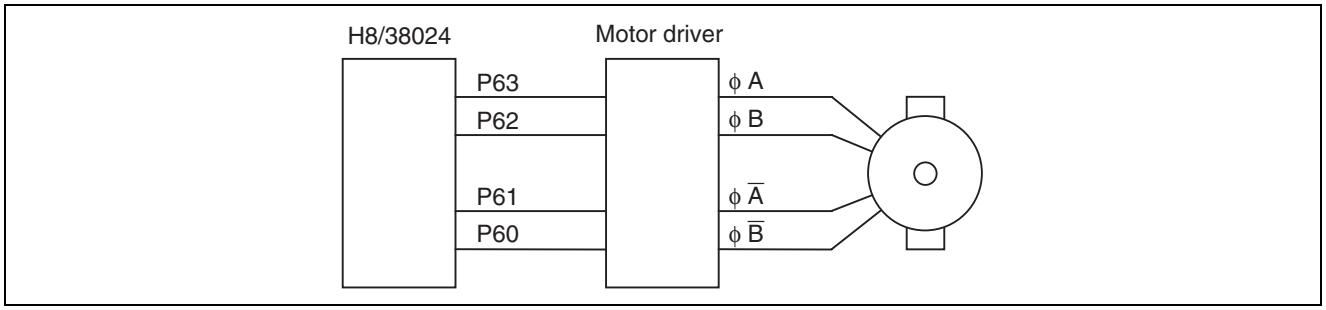


Figure 1 Connections for Controlling a Two-Phase Stepper Motor

2. Description of Functions

2.1 Motor Specifications

This sample task uses a permanent magnet stepper motor (KP6P8-701 manufactured by Japan Servo, Co. Ltd.). Table 1 is a list of the standard specifications of the KP6P8-701.

Table 1 Standard Specifications of the Model KP6P8-701

Item	Value
Phases	2
Stepping angle [deg./step]	7.5
Voltage [V]	12
Current [A/PHASE]	0.33
Resistance of windings [Ω/PHASE]	36
Inductance [mH/PHASE]	28
Maximum static torque [mN·m]	78.4
Detent torque [mN·m]	1.3
Rotor inertia [g·cm^2]	23.7

2.2 Functions

The following describes the H8/38024 functions used for stepper motor control. Figure 2 is a block diagram of the functions used for this sample task.

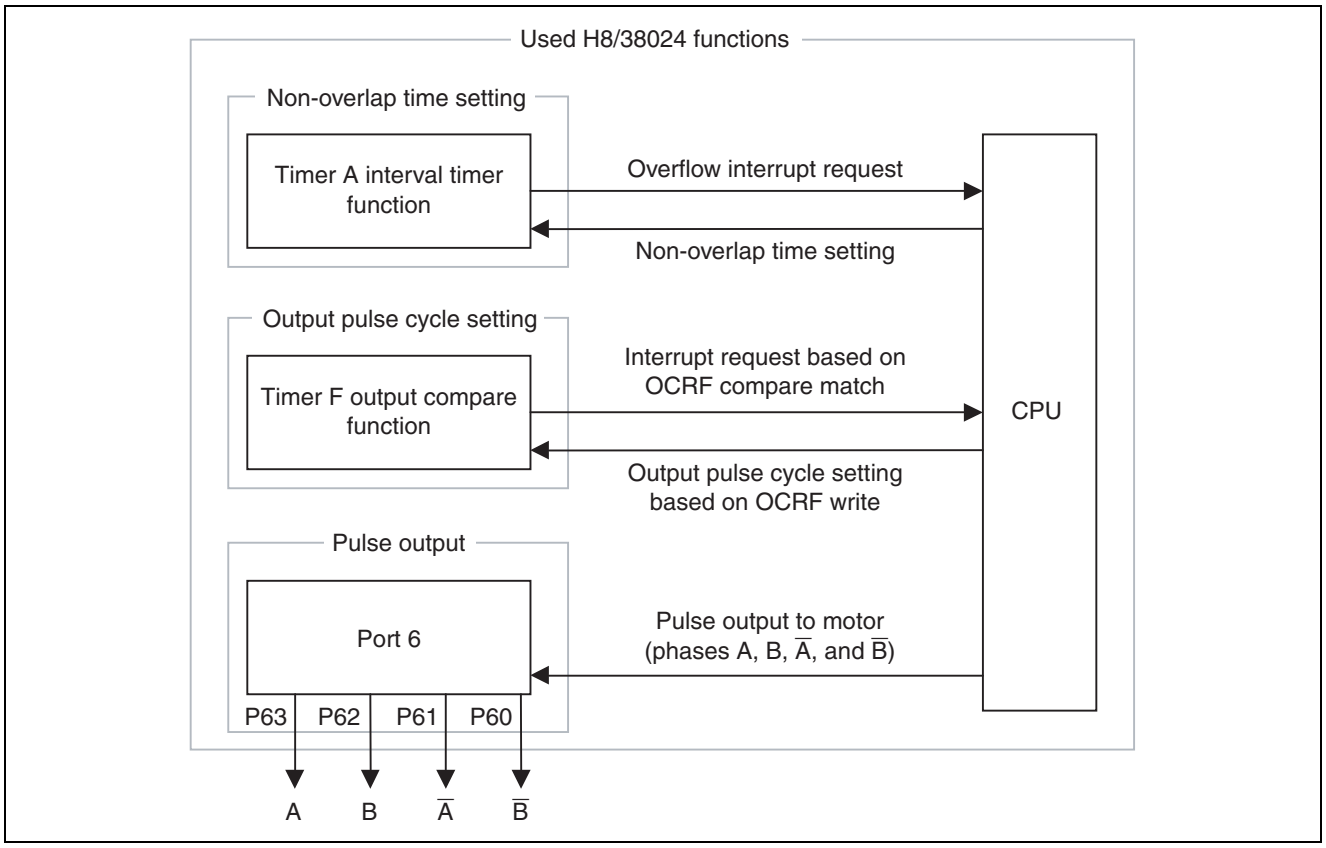


Figure 2 H8/38024 Functions Used

2.3 Timer A Interval Functions

This sample task uses the interval function of timer A. Figure 3 is a block diagram of timer A. This block diagram of timer A is explained below.

- Timer mode register A (TMA)
  - This register is an 8-bit read/write register, and is used to select a prescalar and an input clock. This sample task selects PSS as a prescalar and a prescalar division ratio of 32.
- Timer counter A (TCA)
  - This counter is an 8-bit read up-counter, and is incremented according to an input internal clock. The timer A overflow interrupt request (IRRTA) of interrupt request register 1 (IRR1) is set to 1 when an overflows occurs in TCA.
- Timer A overflow interrupt request flag (IRRTA)
  - This flag is set to 1 when an overflow occurs in TCA. A timer A interrupt is accepted and timer A interrupt processing is started when IRRTA is set to 1, the timer A interrupt enable (IENTA) of interrupt enable register 1 (IENR1) is set to 1, and the I bit of the condition code register (CCR) is cleared to 0.

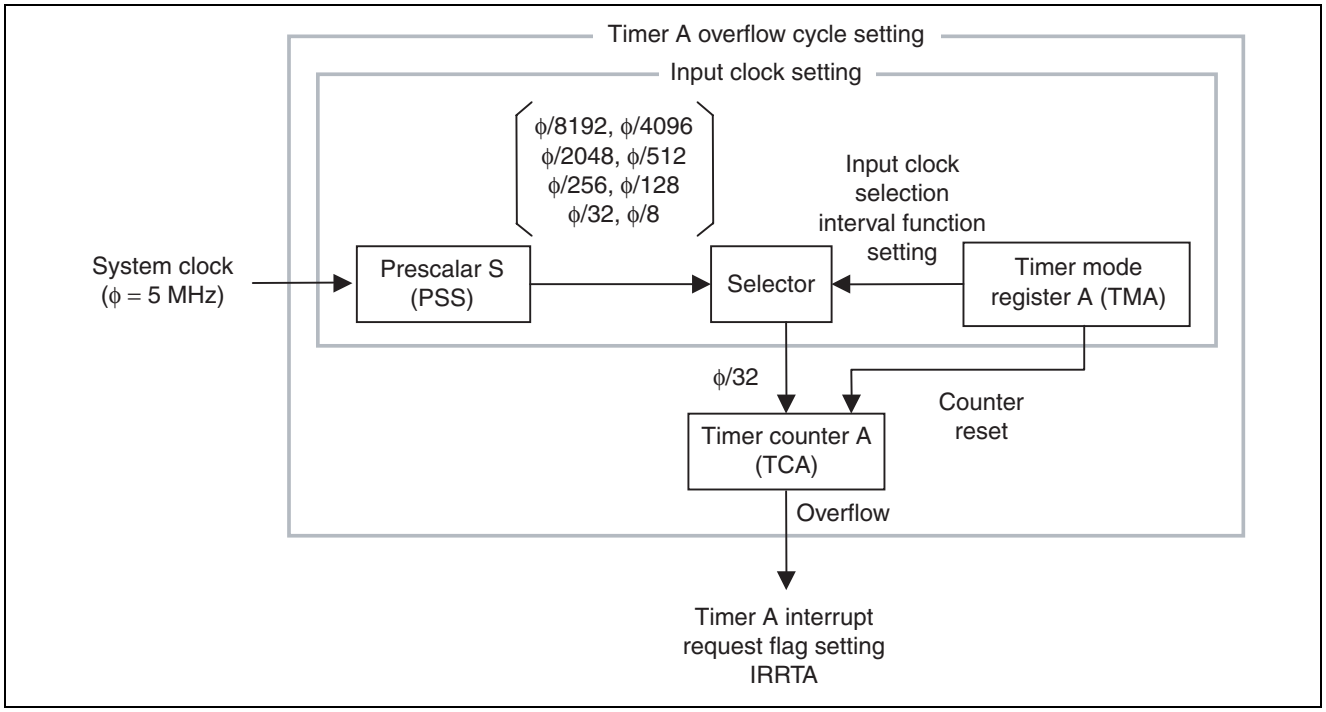


Figure 3 Timer A Interval Function

2.4 Timer F Output Compare Function

Timer F is a 16-bit timer that incorporates an output compare function. This sample task uses the output compare function of timer F. Figure 4 is a block diagram of timer F. This block diagram of timer F is explained below.

- Timer control register F (TCRF)
  - This register is an 8-bit read/write register, and is used to switch between the 16-bit mode and 8-bit mode and select any of four types of internal clocks and an external event.
- Timer control status register F (TCSRF)
  - This register is an 8-bit register, and is used to select counter clearing, set an overflow flag, set a compare match flag, and control whether to enable an interrupt request due to an overflow.
- Timer counter F (TCF) (TCFH, TCFL)
  - This counter is a 16-bit read/write up-counter. This counter is incremented according to an input internal/external clock. As an input clock, one of the following five types can be selected: 4-divided system clock, 16-divided system clock, 32-divided system clock, 4-divided subclock, and external clock. This sample task selects a 4-divided system clock (ϕ/4) as a TCF input clock.
- Output compare register F (OCRF) (OCRFH, OCRFL)
  - This register is a 16-bit read/write register. The contents of OCRF are compared with TCF at all times. When a match is found, a compare match FH occurs, generating an interrupt.

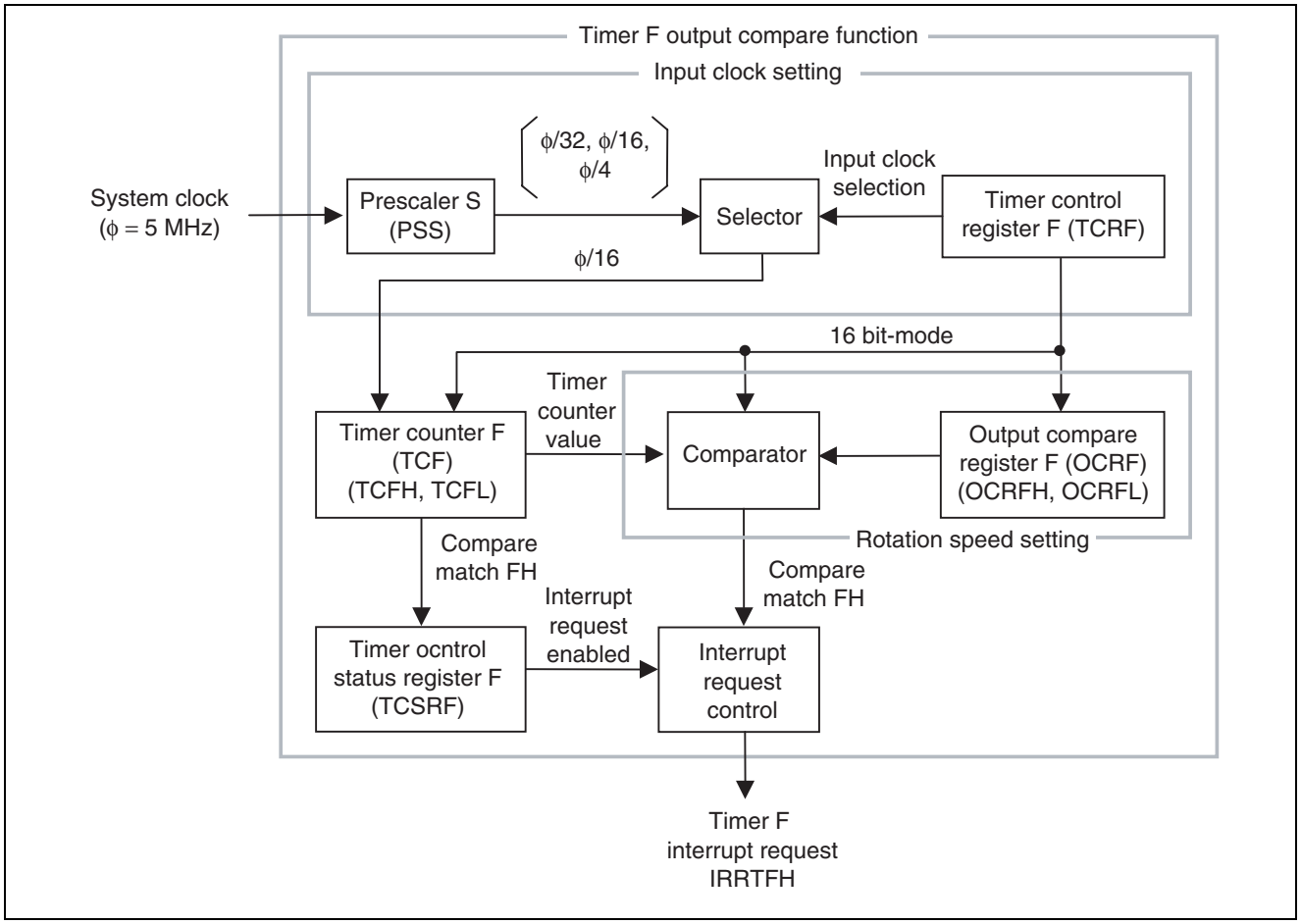


Figure 4 Block Diagram of Timer F

2.5 Port 6 Setting

Port 6 is an 8-bit I/O port. This sample task uses P63 to P60 of port 6. Figure 5 is a block diagram of port 6. The functions of port 6 are explained below.

- Port data register 6 (PDR6)
  — P63 to P60 are used for excitation phase driving of the stepper motor.
- Port control register 6 (PCR6)
  — P63 to P60 are set as output pins.

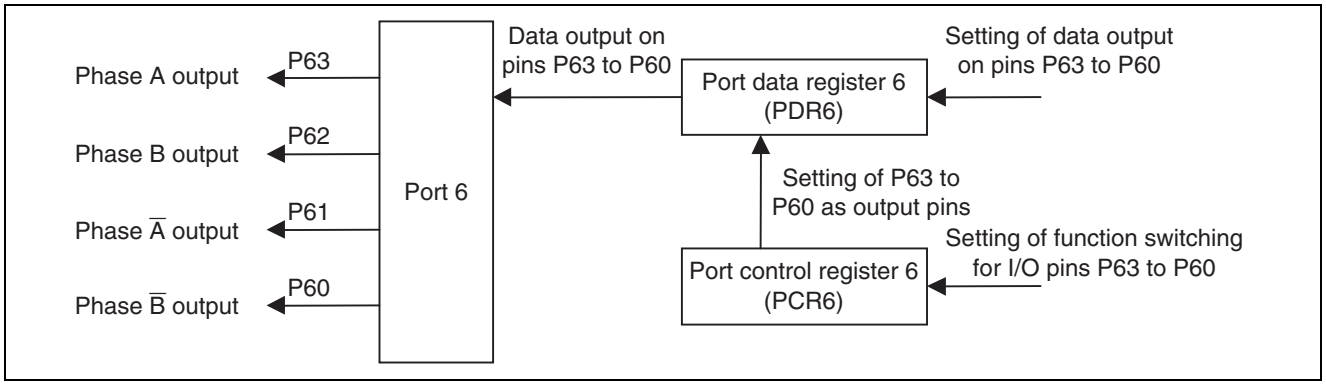


Figure 5 Block Diagram of Port 6 Functions

2.6 Function Assignments

Table 2 is a list of the function assignments for this sample task.

Table 2 Assignment of Functions

Elements	Description
PSS	13-bit up-counter to which system clock (ϕ) is input
TCRF	Sets a TCF input clock, and sets timer F to the 16-bit mode.
TCSRF	Sets a compare match flag, TCF clearing, and a conditions for clearing TCF.
TCF (TCFH, TCFL)	16-bit counter to which a clock of ϕ/16 is input
OCRF (OCRFH, OCRFL)	Sets the duration of one step of the stepper motor.
PDR6	Outputs, from P63 to P60, signals for driving the excitation phase of the stepper motor.
PCR6	Sets P63 to P60 as output pins.
IENTFH	Enables timer FH interrupt requests.
IRRTFH	Timer FH interrupt request flag

3. Principles of Operation

3.1 Example of Stepper Motor Operation

Figure 6 shows an example of operating the two-phase stepper motor with a stepping angle of 7.5 [deg./step] by using two-phase excitation. The operation is outlined below.

- As shown in figure 6, a high pulse causes the corresponding phase to be excited.
- First, phases $\overline{B}$ and A are excited. At this time, the rotor is positioned halfway between phases $\overline{B}$ and A.
- Next, phases A and B are excited simultaneously. At this time, the rotor is positioned halfway between phases A and B. Then, the two-phase excitation method rotates the rotor by exciting two adjacent phases (phases B and $\overline{A}$ → phases $\overline{A}$ and $\overline{B}$ → phases $\overline{B}$ and A → phases A and B).
- For reverse rotation, the stepper motor is rotated by excitation in the following order: phases $\overline{A}$ and $\overline{B}$ → phases B and $\overline{A}$ → phases A and B → phases $\overline{B}$ and A.
- For stop operation, the stepper motor is stopped by keeping the last phase of a forward rotation or reverse rotation excited for a certain period of time.

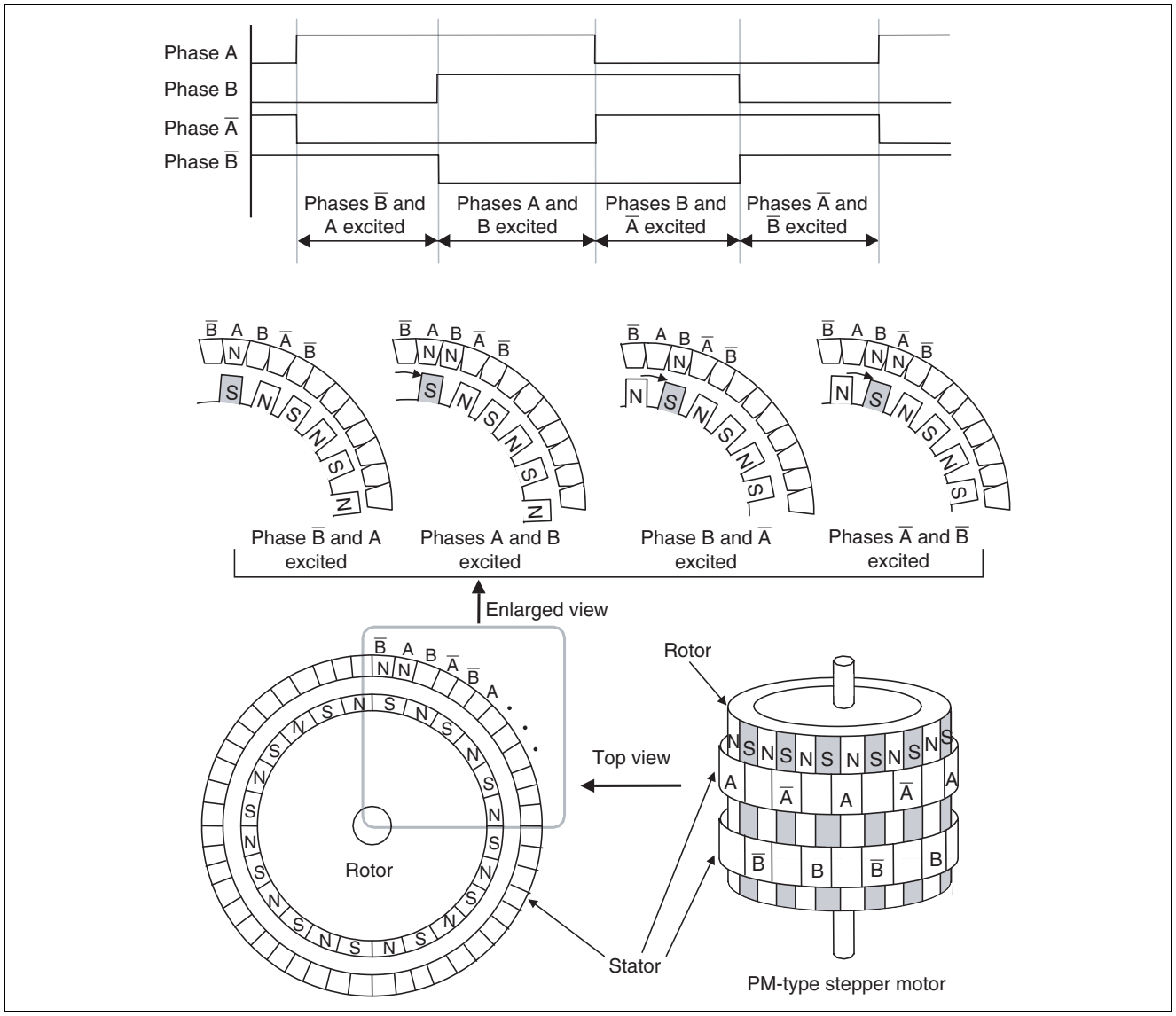


Figure 6 Example of Stepper Motor Operation

3.2 Non-Overlap Time

As part of output pattern switching, a through-current protection period n (non-overlap time) is inserted. The turn-off delay that occurs upon excitation phase switching can destroy a driver. To prevent this, a non-overlap time is inserted to allow for the time delay.

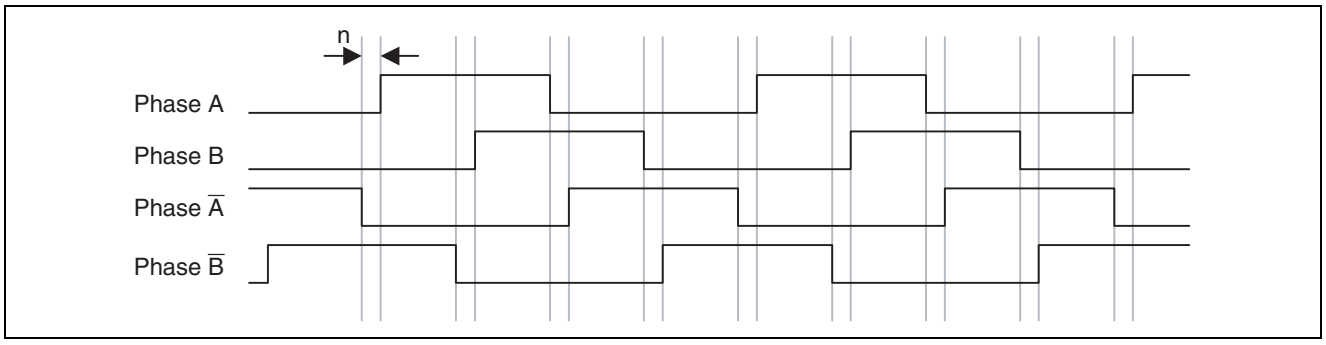


Figure 7 Example of Non-Overlap Time Output

3.3 Slew-up and Slew-down Operation

Slew-up/slew-down operation maintains the synchronization of the motor. Out-of-synchronization means that if a series of short-cycle pulses are suddenly output to operate the motor, the motor may not be able to handle the load and will not rotate. Slew-up and slew-down operation is used to avoid this problem. The following explains the principle of the operation.

- The pulse cycles are gradually shortened to output the specified number of pulses (slew-up operation).
- The specified number of pulses are output at a regular pulse cycle (constant-speed operation).
- The pulse cycle is gradually extended to output the specified number of pulses (slew-down operation).

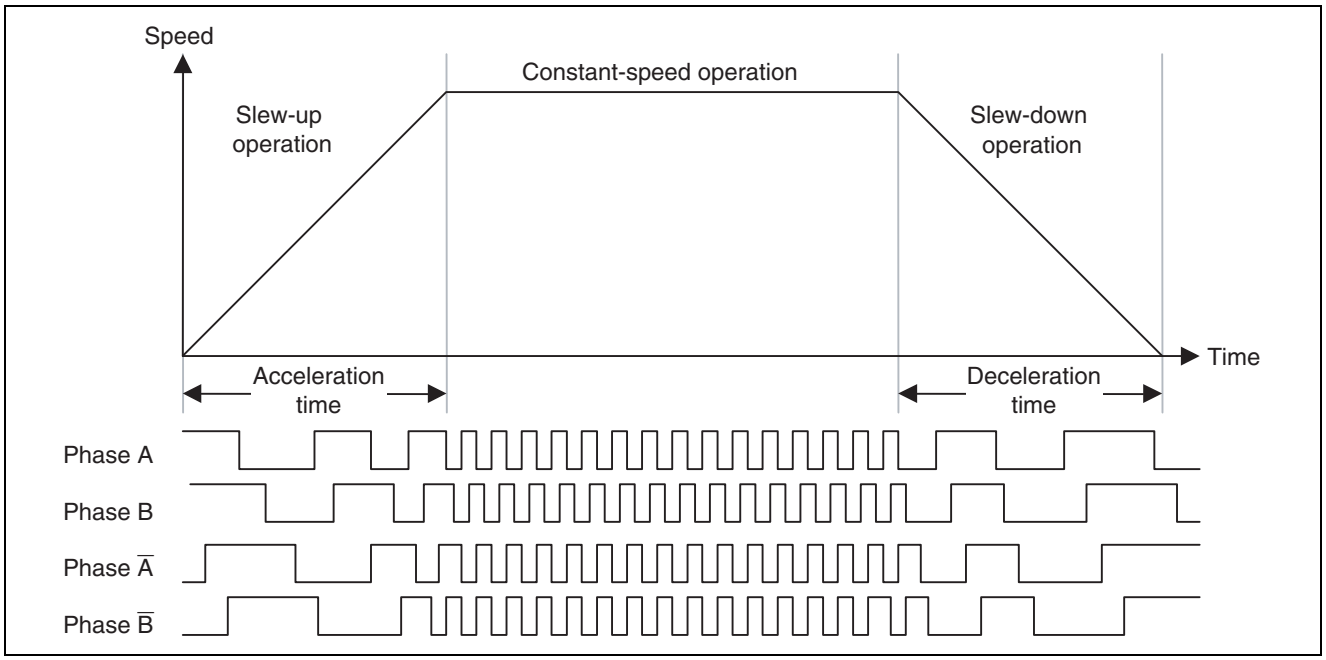


Figure 8 Example of Slew-up and Slew-down Operation

3.4 Flowchart

Figure 9 is a flowchart illustrating stepper motor control.

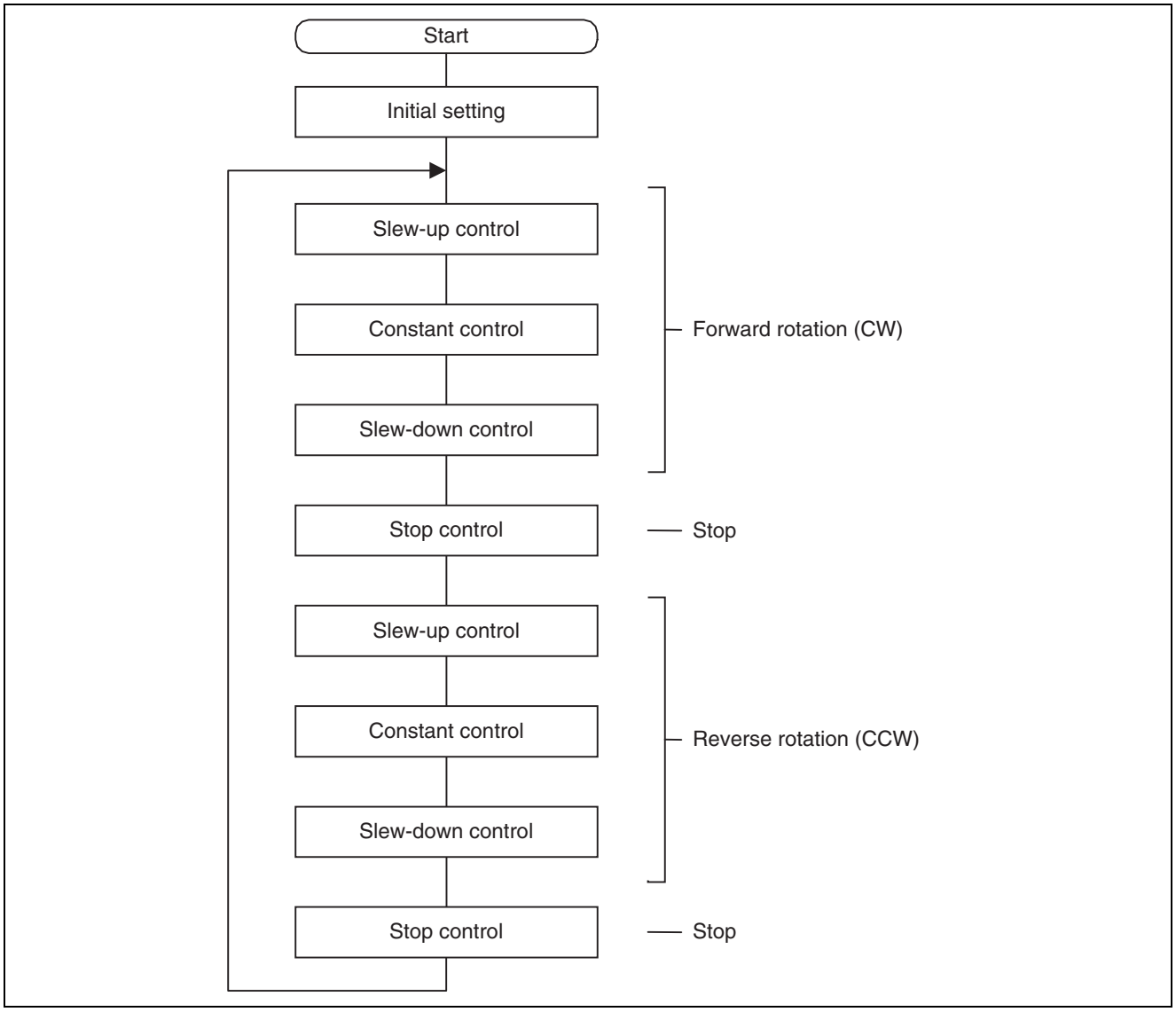


Figure 9 Flowchart of Stepper Motor Control

3.5 Expression for Calculating Timer A Overflow Time (Non-Overlap Time)

- The expression for calculating a TCA overflow cycle is as follows:

$$\text{TCA overflow cycle} = \frac{1}{\text{System clock /32}} \times 256$$
$$= 1.64\text{ms}$$

3.6 Expression for Calculating Timer F Interrupt Time

- Setting the output compare register (OCRF) enables the timer F interrupt time to be calculated as shown below.

$$\text{Timer F interrupt time} = \frac{\text{OCRF + 1}}{\text{(System clock } \phi\text{/16)}}$$
$$= \frac{\text{OCRF + 1}}{\text{(5MHz/16)}}$$
$$= 3.2 \times \text{(OCRF + 1)} \quad [\mu\text{s}]$$

3.7 Slew-up Control during Forward Rotation

Figure 10 illustrates the principle of slew-up control during forward rotation.

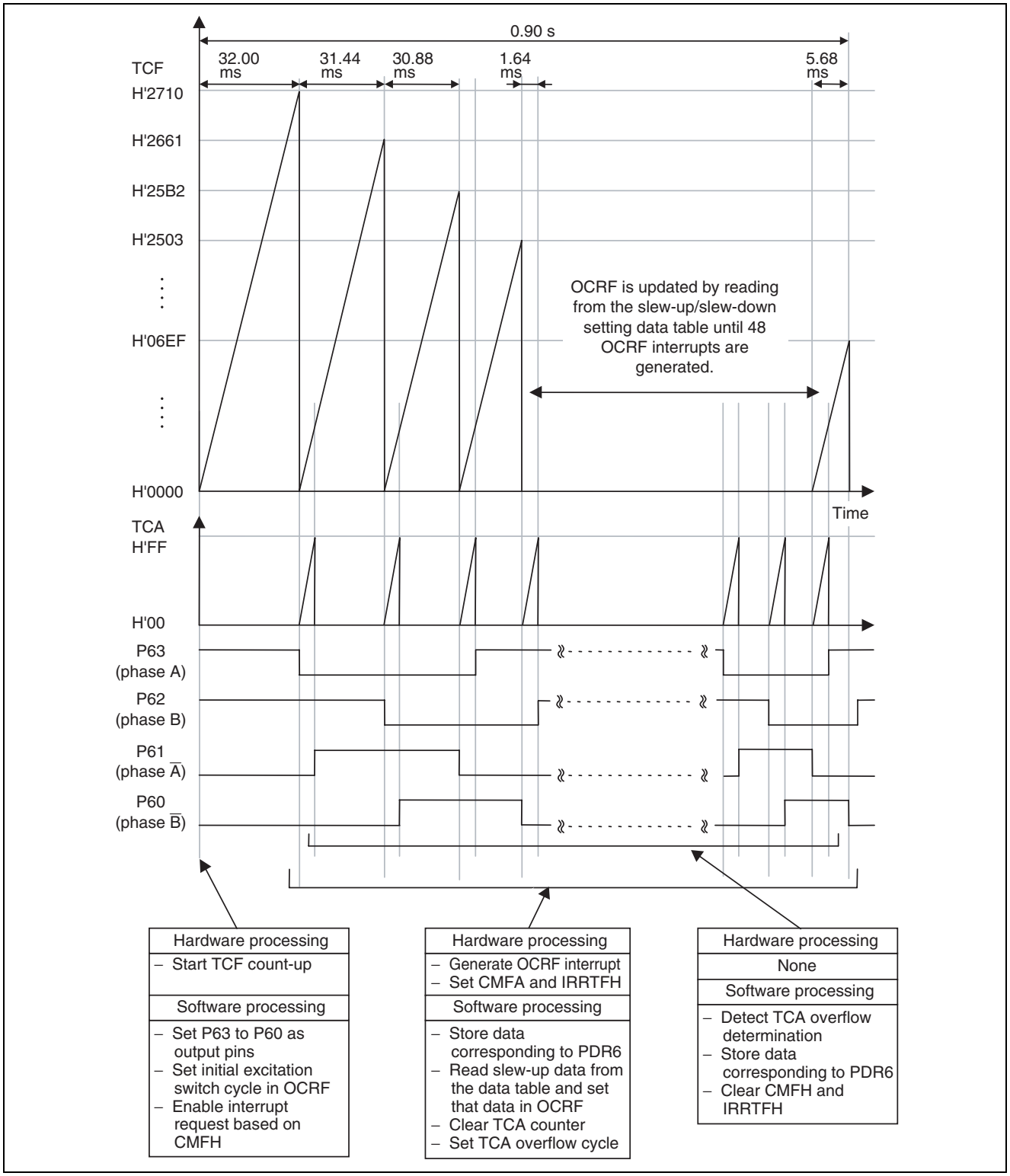


Figure 10 Principle of Slew-up Control during Forward Rotation

3.8 Constant Control during Forward Rotation

Figure 11 illustrates the principle of constant control during forward rotation.

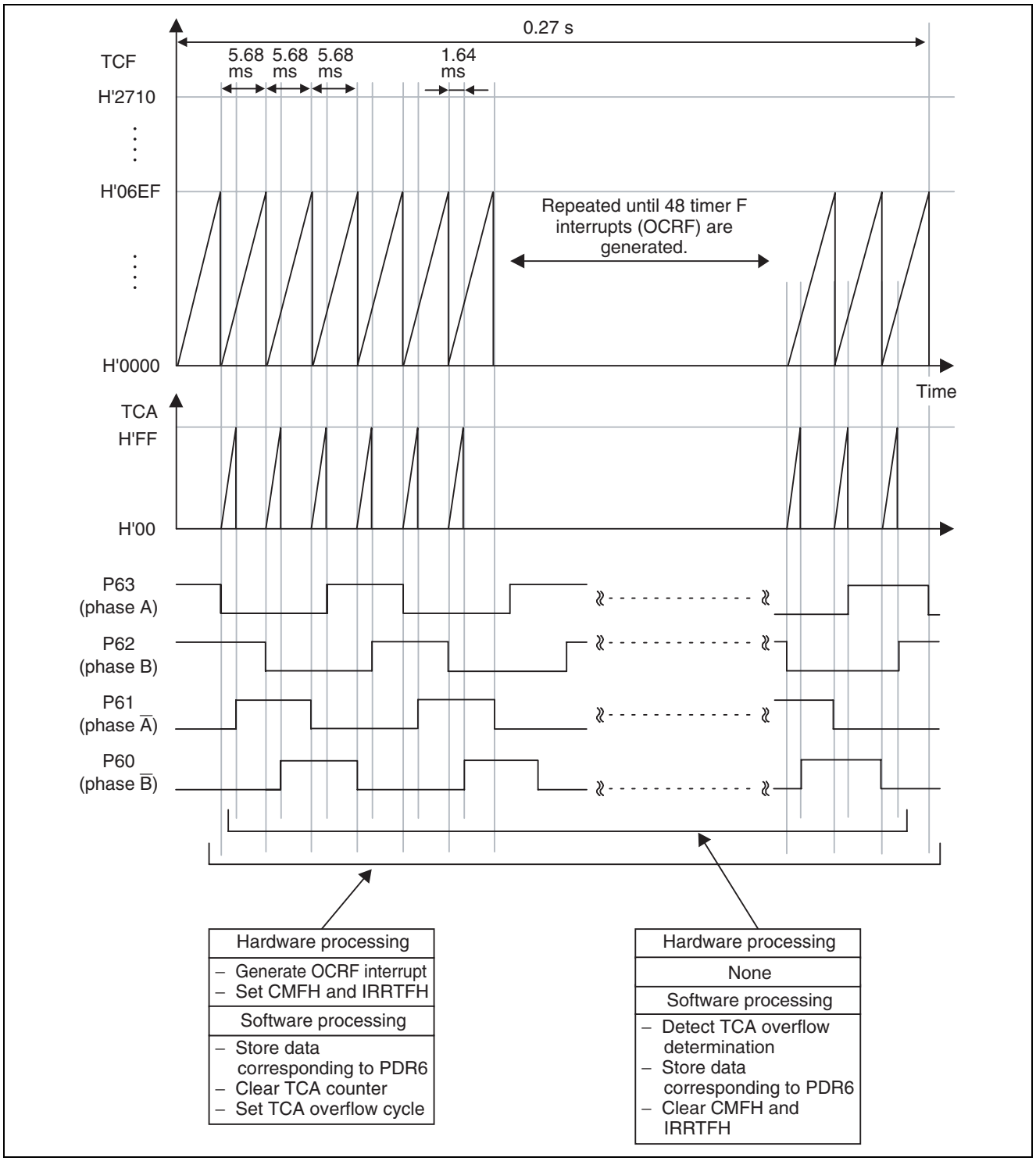


Figure 11 Principle of Constant Control during Forward Rotation

3.9 Slew-down Control during Forward Rotation

Figure 12 illustrates the principle of slew-down control during forward rotation.

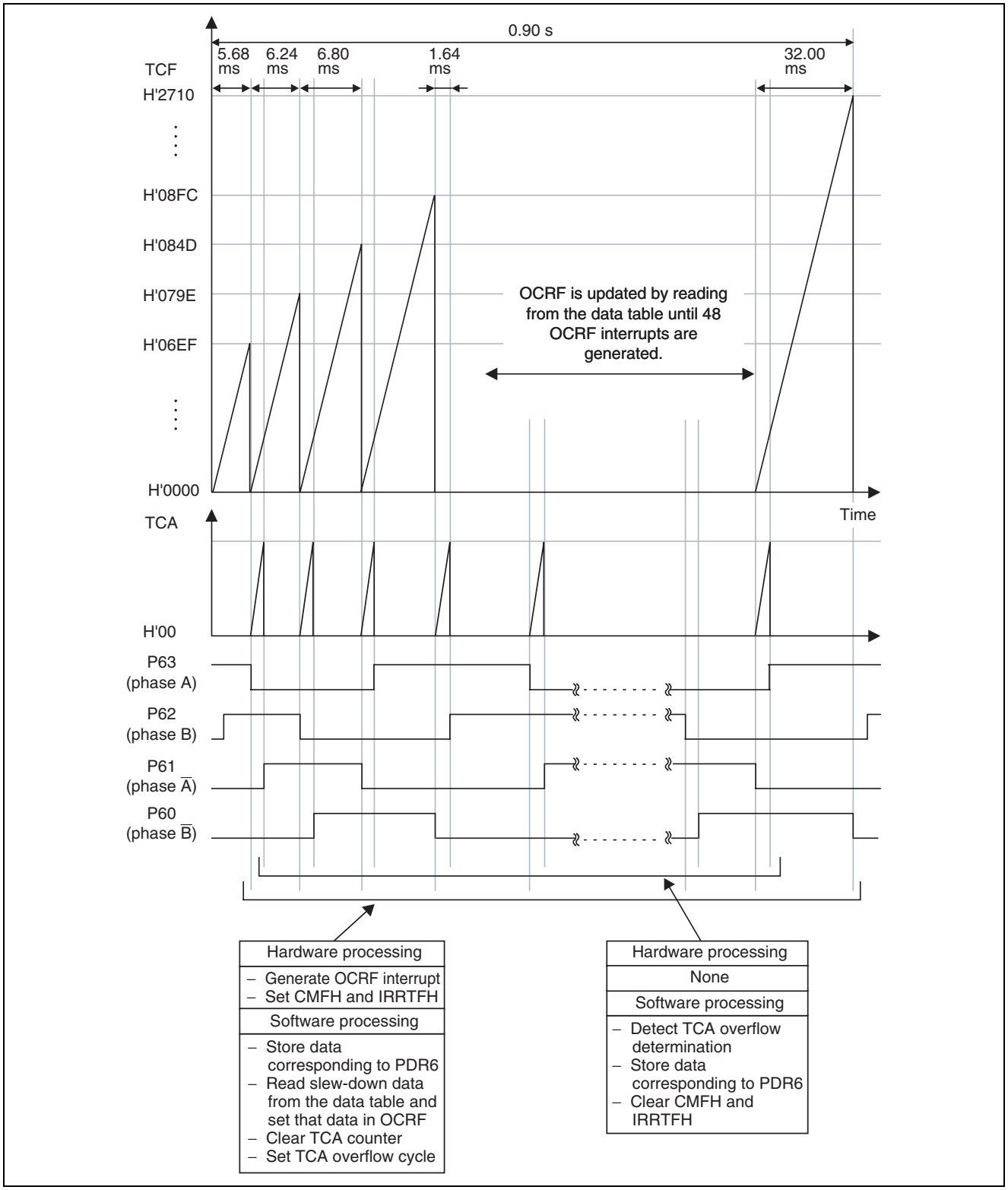


Figure 12 Principle of Slew-down Control during Forward Rotation

3.10 Stop Control

Figure 13 illustrates the principle of stop control.

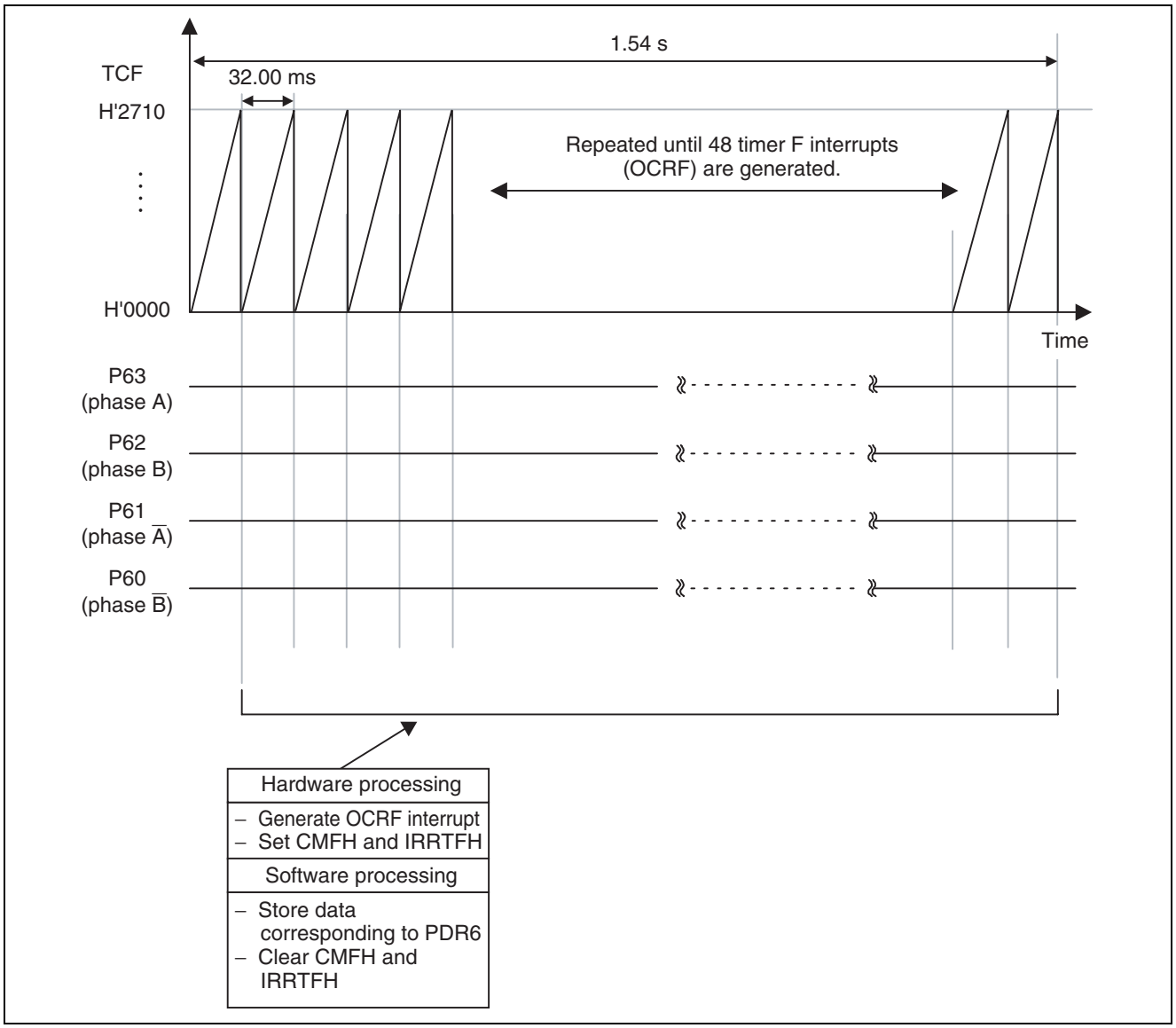


Figure 13 Principle of Stop Control

3.11 Slew-up Control during Reverse Rotation

Figure 14 illustrates the principle of slew-up control during reverse rotation.

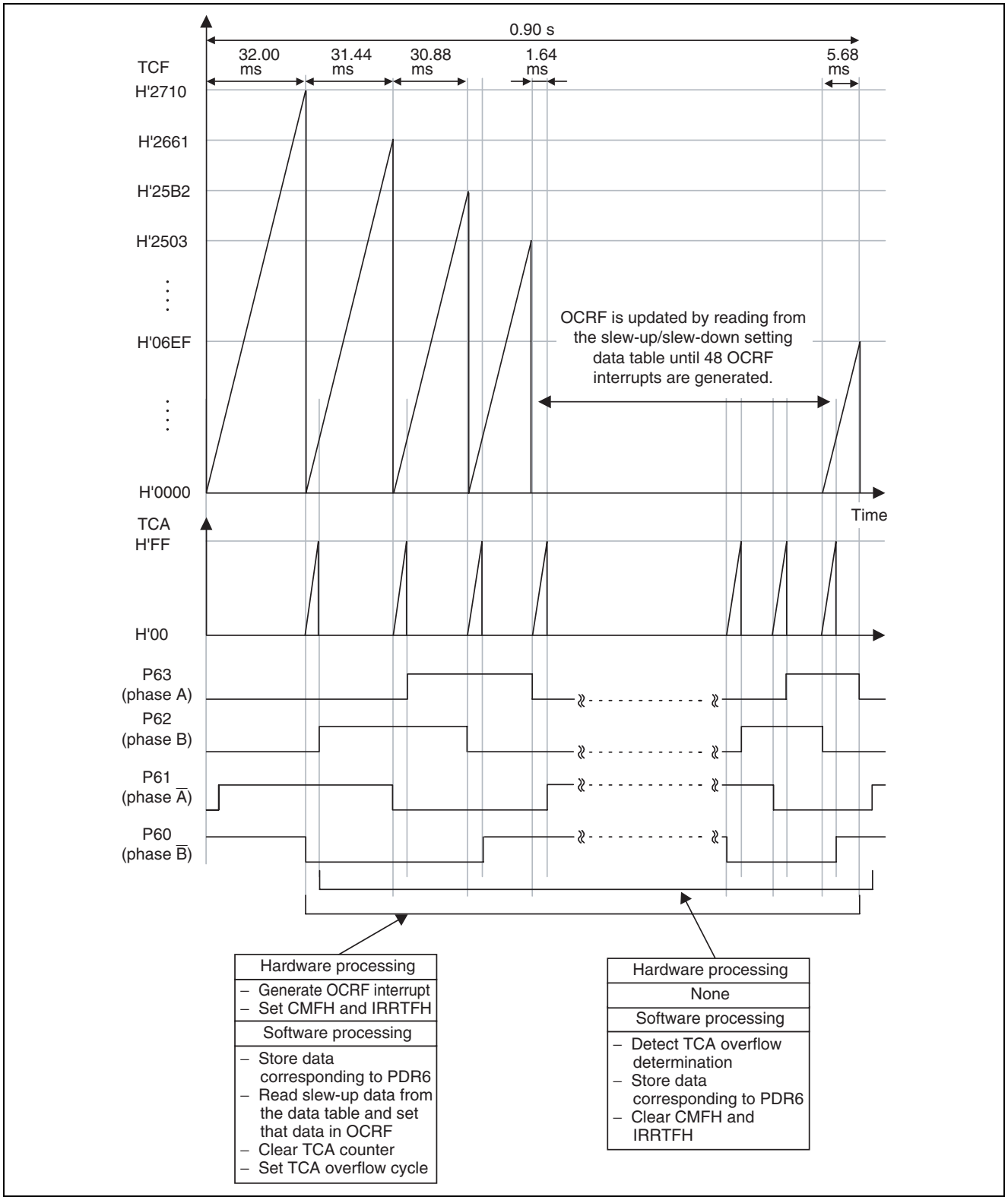


Figure 14 Principle of Slew-up Control during Reverse Rotation

3.12 Constant Control during Reverse Rotation

Figure 15 illustrates the principle of constant control during reverse rotation.

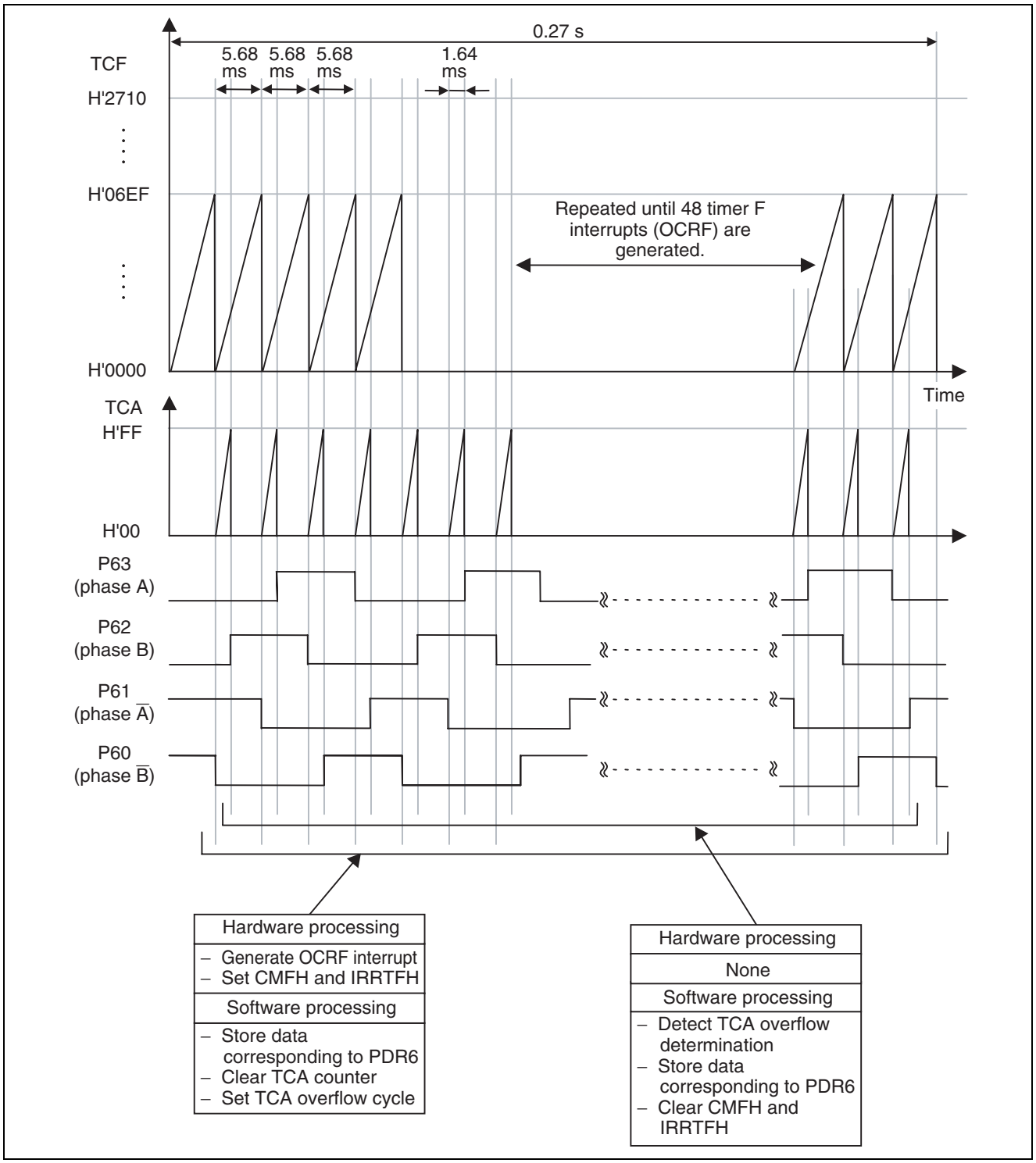


Figure 15 Principle of Constant Control during Reverse Rotation

3.13 Slew-down Control during Reverse Rotation

Figure 16 illustrates the principle of slew-down control during reverse rotation.

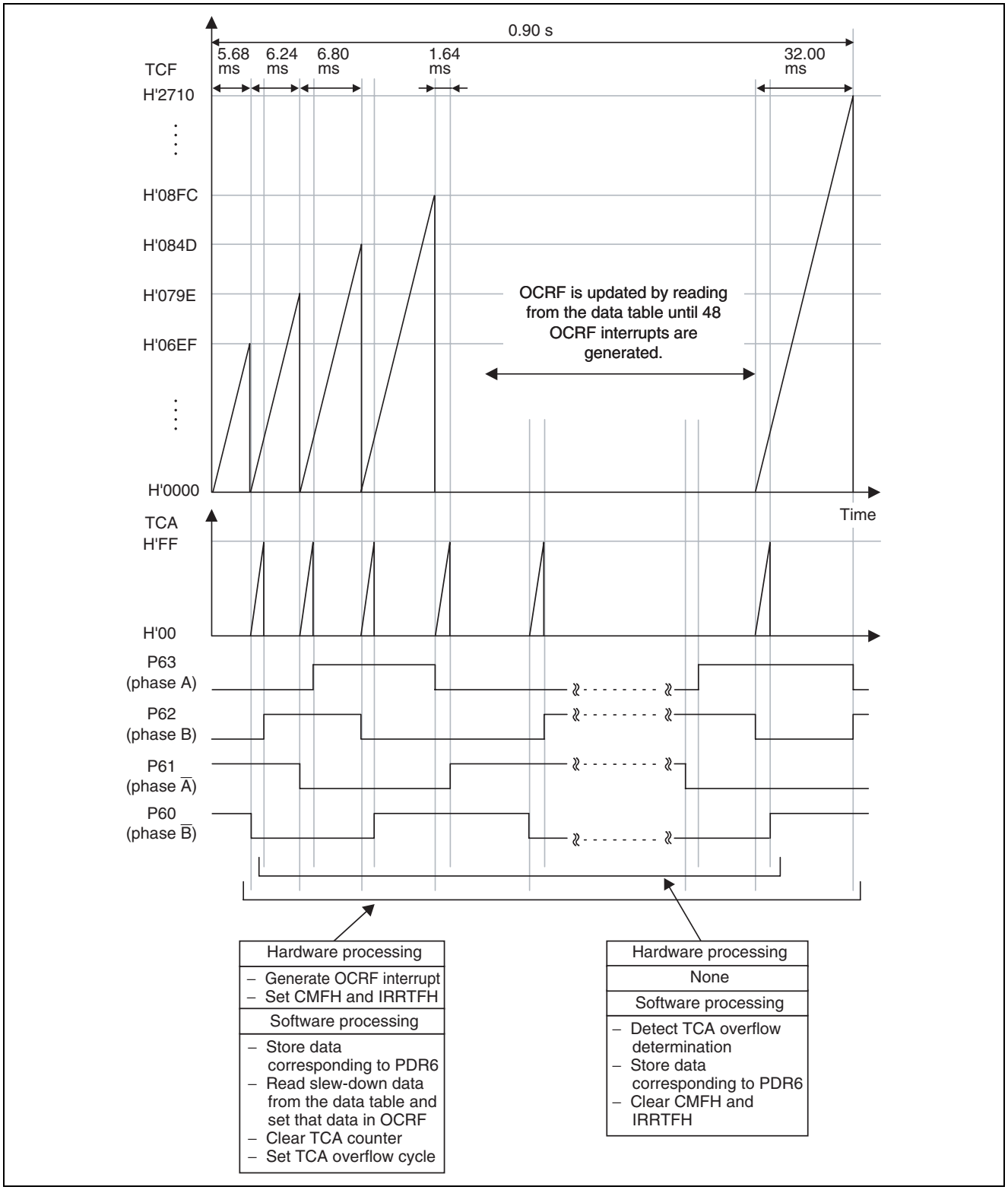


Figure 16 Principle of Slew-down Control during Reverse Rotation

4. Description of Software

4.1 Modules

Table 3 is a list of the modules used for this sample task. Figure 17 shows the hierarchical structure of this sample task.

Table 3 Modules

Label Name	Function
main	Main routine Initializes the global variables, I/O ports, and timer F, and enables interrupts.
tfhint	Timer FH interrupt processing Main routine for the stepper motor
fslueup	Slew-up control during forward rotation
fsluedwn	Slew-down control during forward rotation
fconst	Constant control during forward rotation
frstop	Forward/reverse rotation stop
rslueup	Slew-up control during reverse rotation
rsluedwn	Slew-down control during reverse rotation
rconst	Constant control during reverse rotation
n_overlap	Non-overlap time setting

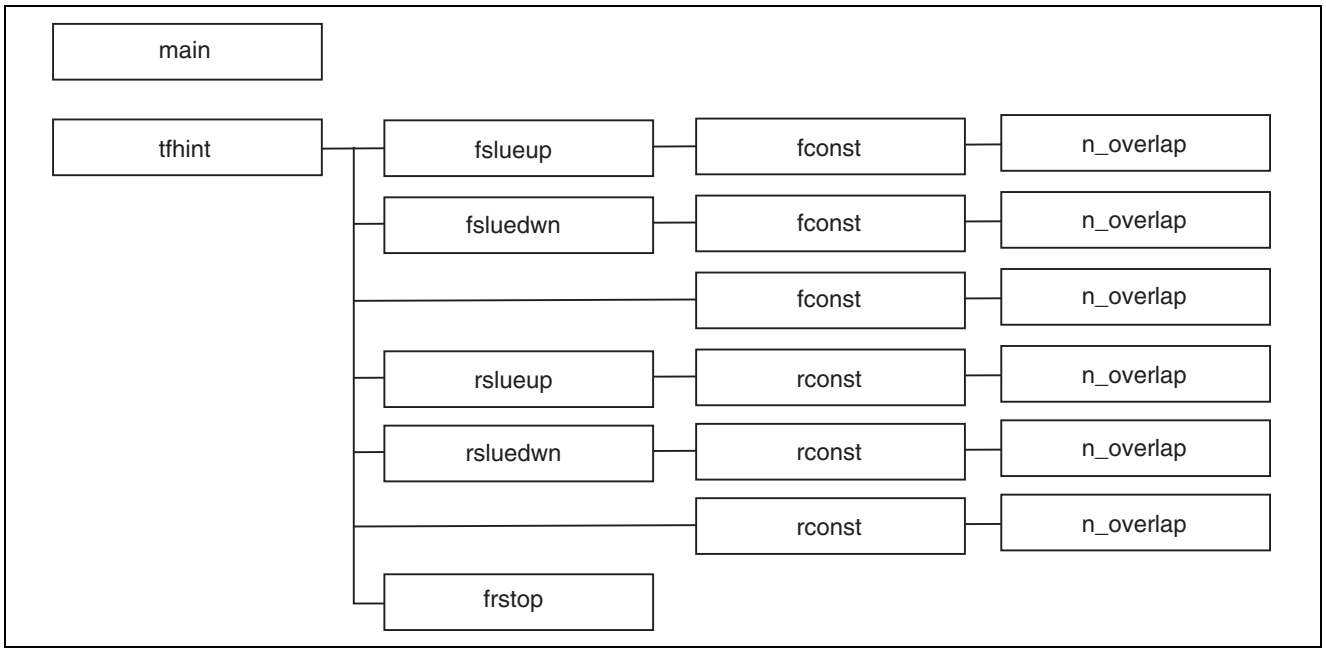


Figure 17 Hierarchical Structure

4.2 Data Table Variables

- Data table for switching the excitation pattern of the stepper motor

```
pattbl[8] = {
   0x08,  ....Excites phase A (P63).
   0x0C,  ....Excites phases A (P63) and B (P62).
   0x04,  ....Excites phase B (P62).
   0x06,  ....Excites phases B (P62) and Ā (P61).
   0x02,  ....Excites phase Ā (P61).
   0x03,  ....Excites phases Ā (P61) and B̄ (P60).
   0x01,  ....Excites phase B̄ (P60).
   0x09,  ....Excites phases B̄ (P60) and A (P63).
};
```

- Data table for slew-up and slew-down setting

```
uptbl[48] = {
   0x2710,0x2661,0x25B2,0x2503,0x2454,0x23A5,0x22F6,0x2247,0x2198,0x20E9,
   0x203A,0x1F8B,0x1EDC,0x1E2D,0x1D7E,0x1CCF,0x1C20,0x1B71,0x1AC2,0x1A13,
   0x1964,0x18B5,0x1806,0x1757,0x16A8,0x15F9,0x154A,0x149B,0x13EC,0x133D,
   0x128E,0x11DF,0x1130,0x1081,0x0FD2,0x0F23,0x0E74,0x0DC5,0x0D16,0x0C67,
   0x0BB8,0x0B09,0x0A5A,0x09AB,0x08FC,0x084D,0x079E,0x06EF
};
```

Data in uptbl[] is sequentially written to OCRF by an OCRF interrupt generated during slew-up and slew-down until the stepper motor makes one complete revolution (48 steps).

4.3 Description of Module

4.3.1 main

1. Module specifications
   Function overview: Initializes the global variables, I/O ports, and timer F, and enables interrupts.

Table 4 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	tcnt	Elements of array pattbl[] representing stepper motor excitation data
	unsigned char	sluecnt	Elements of array uptbl[] used for slew-up and slew-down operation
	unsigned char	nextmode	Sets the operating mode of the stepper motor. 0: Slew-up control during forward rotation 1: Constant control during forward rotation 2: Slew-down control during forward rotation 3: Stop control 4: Slew-up control during reverse rotation 5: Constant control during reverse rotation 6: Slew-down control during reverse rotation 7: Stop control
	unsigned short	modecnt	Sets the number of interrupts in the operating mode of the stepper motor.
ROM	unsigned char	pattbl[8]	Excitation pattern data table for the stepper motor
	unsigned short	uptbl[48]	Interrupt time data table for slew-up and slew-down operation

2. Internal registers
   The internal registers used for this sample task are described below.

- TCRF Timer control register F Address: H′FFB6

Bit	Bit Name	Setting	Function
6	CKSH2	0	Clock select H
5	CKSH1	0	CKSH2 to CKSH0 = B′000, B′001, B′010: Causes TCF to operate as a 16-bit counter.
4	CKSH0	0	
2	CKSL2	1	Clock select L
1	CKSL1	0	CKSL2 to CKSL0 = B′101: Increments TCF with an internal clock of ϕ/16.
0	CKSL0	1	

- TCSRF Timer control status register F Address: H′FFB7

Bit	Bit Name	Setting	Function
6	CMFH	0	Compare match flag H CMFH = 0: No compare match F has occurred. CMFH = 1: A compare match F has occurred.
4	CCLRH	1	CCLRH = 0: Disables TCF clearing based on a compare match in the 16-bit mode. CCLRH = 1: Enables TCF clearing based on a compare match in the 16-bit mode.

- TCF 16-bit timer counter F Address: H′FFB8
  Function: 16-bit counter to which ϕ/16 is input
  Setting: H′0000

- OCRF 16-bit output compare register F Address: H′FFBA
  Function: A compare match occurs when a match is found between the setting of OCRF and the count value of TCF.
  Setting: H′2710

- PDR6 Port data register 6 Address: H′FFD9
  Function: Uses P63 to P60 for excitation phase driving of the stepper motor.
  Setting: H′08

- PCR6 Port control register 6 Address: H′FFE9
  Function: Sets P63 to P60 as output pins when PCR6 = H′0F.
  Setting: H′0F

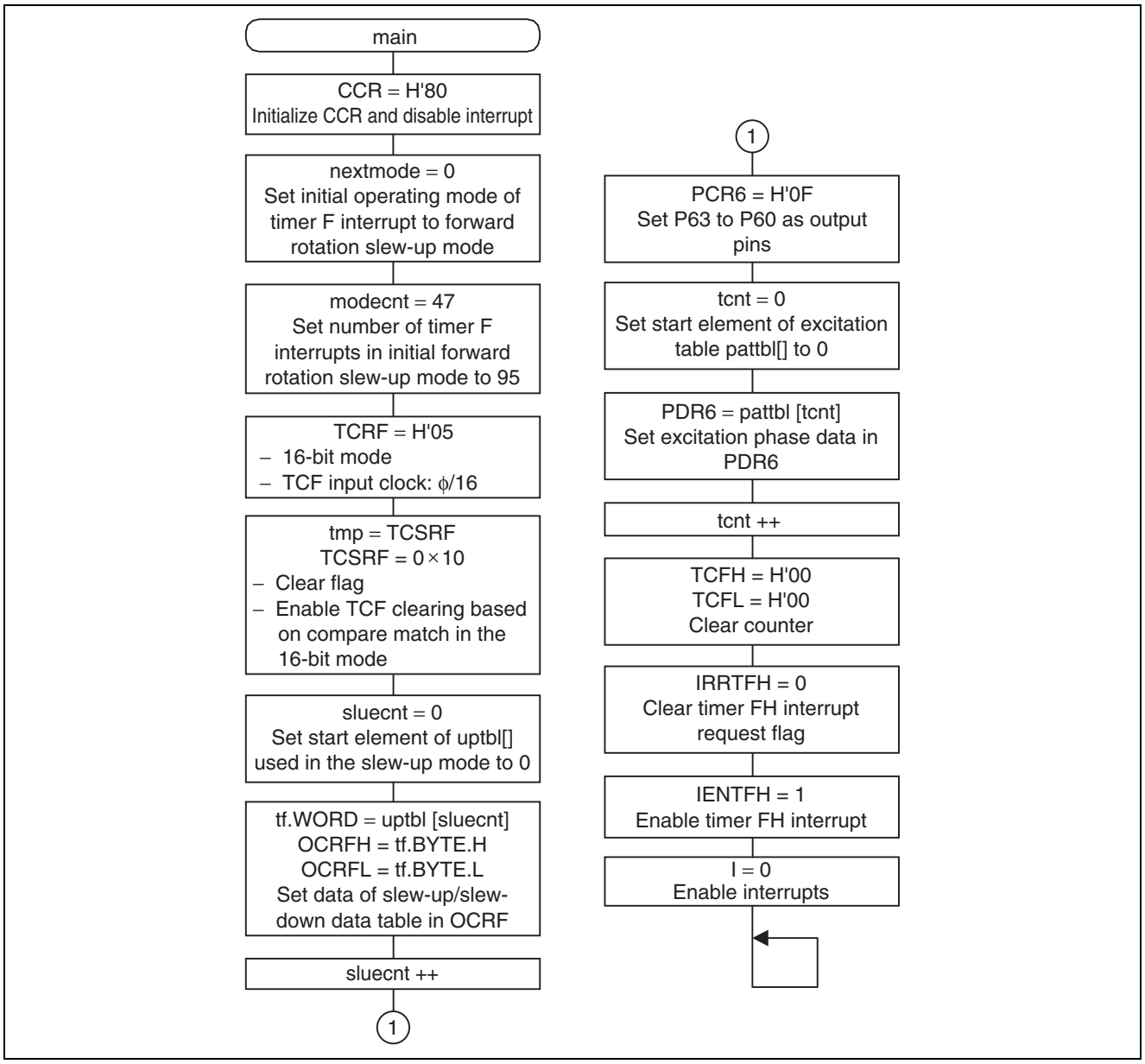
- IENR2 Interrupt enable register 2 Address: H′FFF4

Bit	Bit Name	Setting	Function
3	IENTFH	1	Timer FH interrupt enable IENTFH = 0: Disables timer FH interrupt requests. IENTFH = 1: Enables timer FH interrupt requests.

- IRR2 Interrupt request register 2 Address: H′FFF7

Bit	Bit Name	Setting	Function
3	IRRTFH	0	Timer FH interrupt request flag IRRTFH = 0: No timer FH interrupt request has been made. IRRTFH = 1: A timer FH interrupt request has been made.

3. Flowchart


main
CCR = H'80
Initialize CCR and disable interrupt
nextmode = 0
Set initial operating mode of timer F interrupt to forward rotation slew-up mode
modecnt = 47
Set number of timer F interrupts in initial forward rotation slew-up mode to 95
TCRF = H'05
– 16-bit mode
– TCF input clock: ϕ/16
tmp = TCSRF
TCSRF = 0×10
– Clear flag
– Enable TCF clearing based on compare match in the 16-bit mode
sluecnt = 0
Set start element of uptbl[] used in the slew-up mode to 0
tf.WORD = uptbl [sluecnt]
OCRFH = tf.BYTE.H
OCRFL = tf.BYTE.L
Set data of slew-up/slew-down data table in OCRF
sluecnt ++
1
1
PCR6 = H'0F
Set P63 to P60 as output pins
tcnt = 0
Set start element of excitation table pattbl[] to 0
PDR6 = pattbl [tcnt]
Set excitation phase data in PDR6
tcnt ++
TCFH = H'00
TCFL = H'00
Clear counter
IRRTFH = 0
Clear timer FH interrupt request flag
IENTFH = 1
Enable timer FH interrupt
I = 0
Enable interrupts

4.3.2 tfhint

1. Module specifications
   Function overview: Timer FH interrupt processing/main processing for the stepper motor

Table 5 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	tcnt	Elements of array pattbl[] representing stepper motor excitation data
	unsigned char	sluecnt	Elements of array uptbl[] used for slew-up and slew-down operation
	unsigned char	nextmode	Sets the operating mode of the stepper motor. 0: Slew-up control during forward rotation 1: Constant control during forward rotation 2: Slew-down control during forward rotation 3: Stop control 4: Slew-up control during reverse rotation 5: Constant control during reverse rotation 6: Slew-down control during reverse rotation 7: Stop control
	unsigned short	modecnt	Sets the number of interrupts in the operating mode of the stepper motor.

2. Internal registers
   The internal registers used for this sample task are described below.

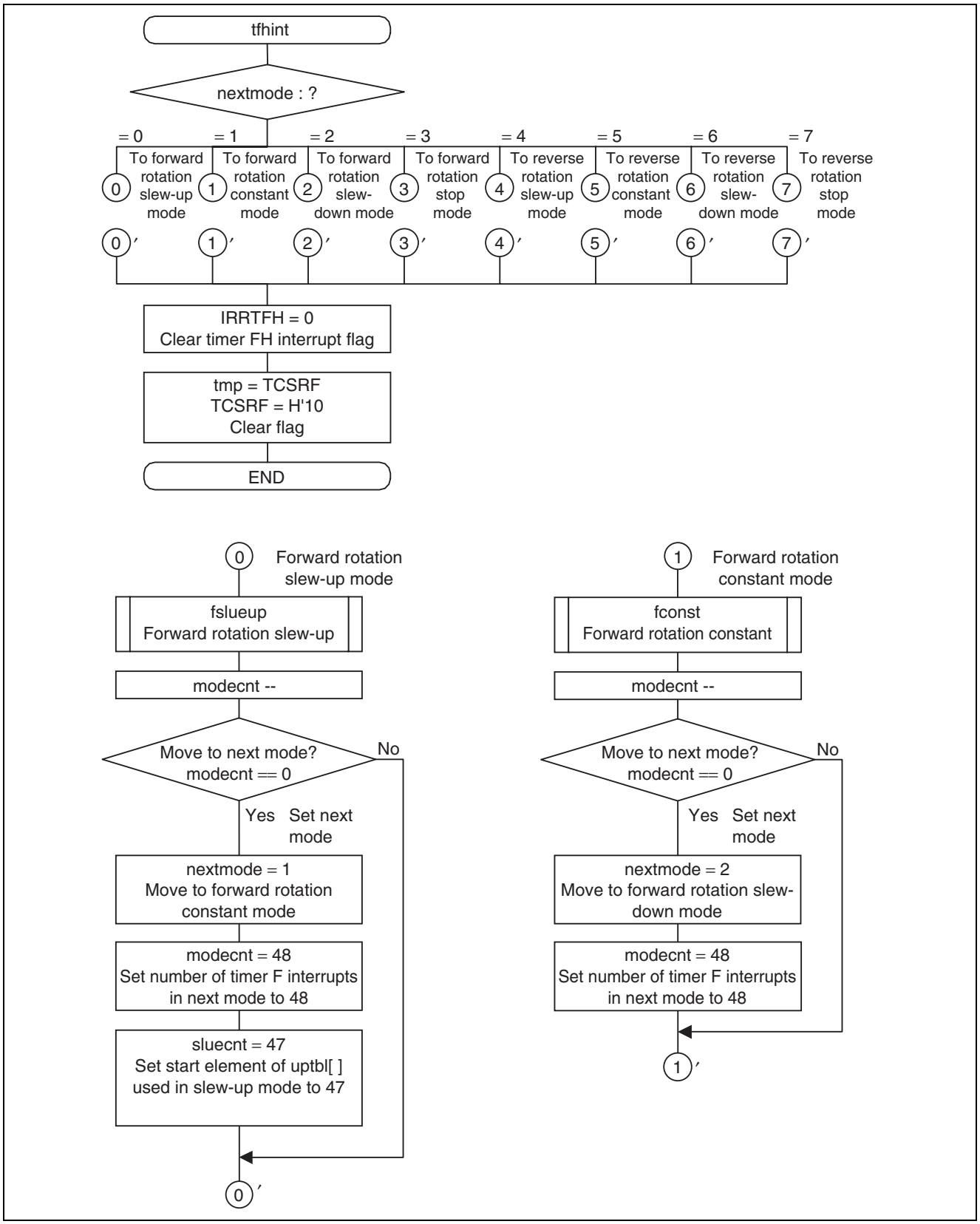
- TCSRF Timer control status register F Address: H′FFB7

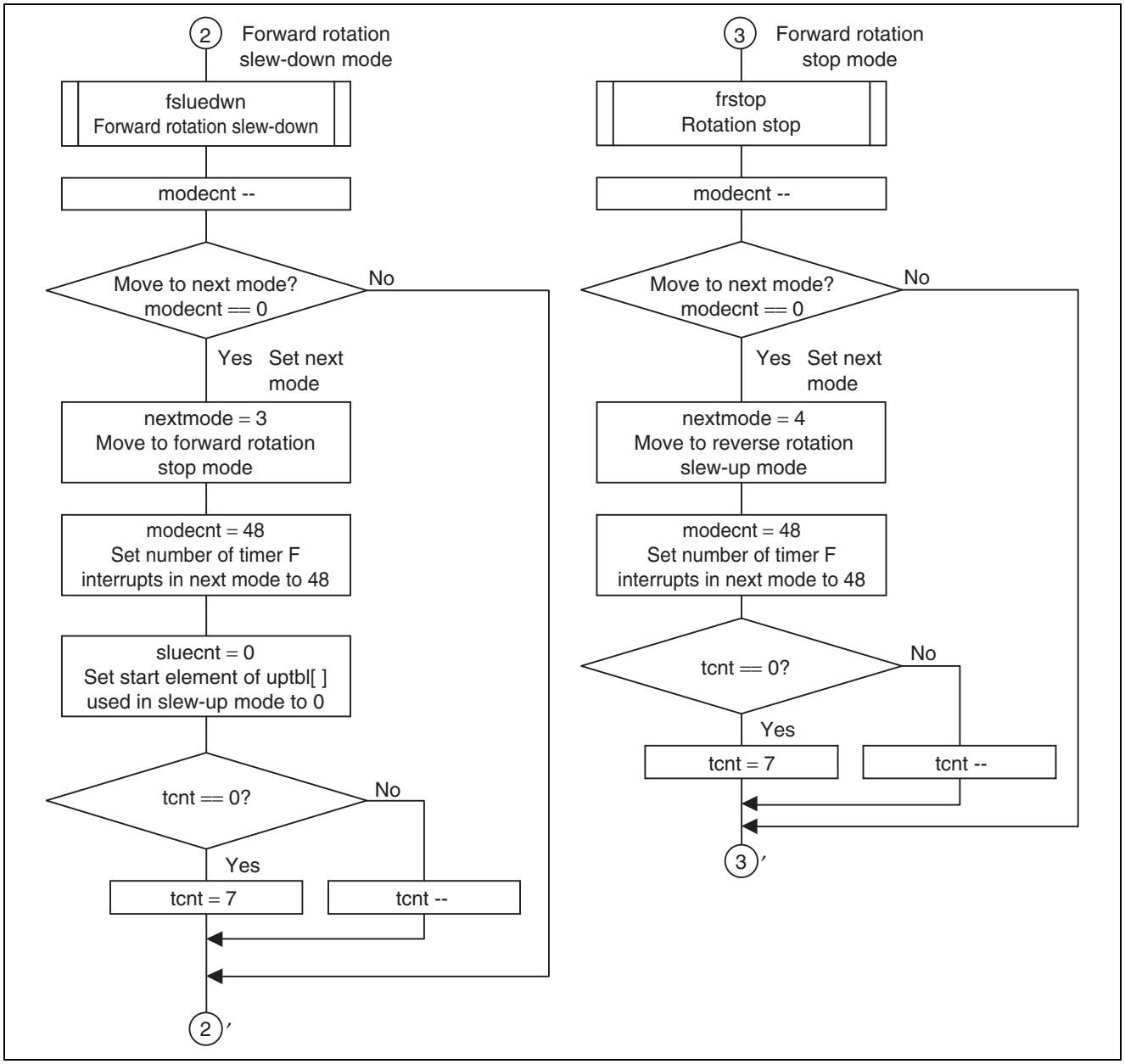
Bit	Bit Name	Setting	Function
6	CMFH	0	Compare match flag H CMFH = 0: No compare match F has occurred. CMFH = 1: A compare match F has occurred.
4	CCLRH	1	CCLRH = 0: Disables TCF clearing based on a compare match in the 16-bit mode. CCLRH = 1: Enables TCF clearing based on a compare match in the 16-bit mode.

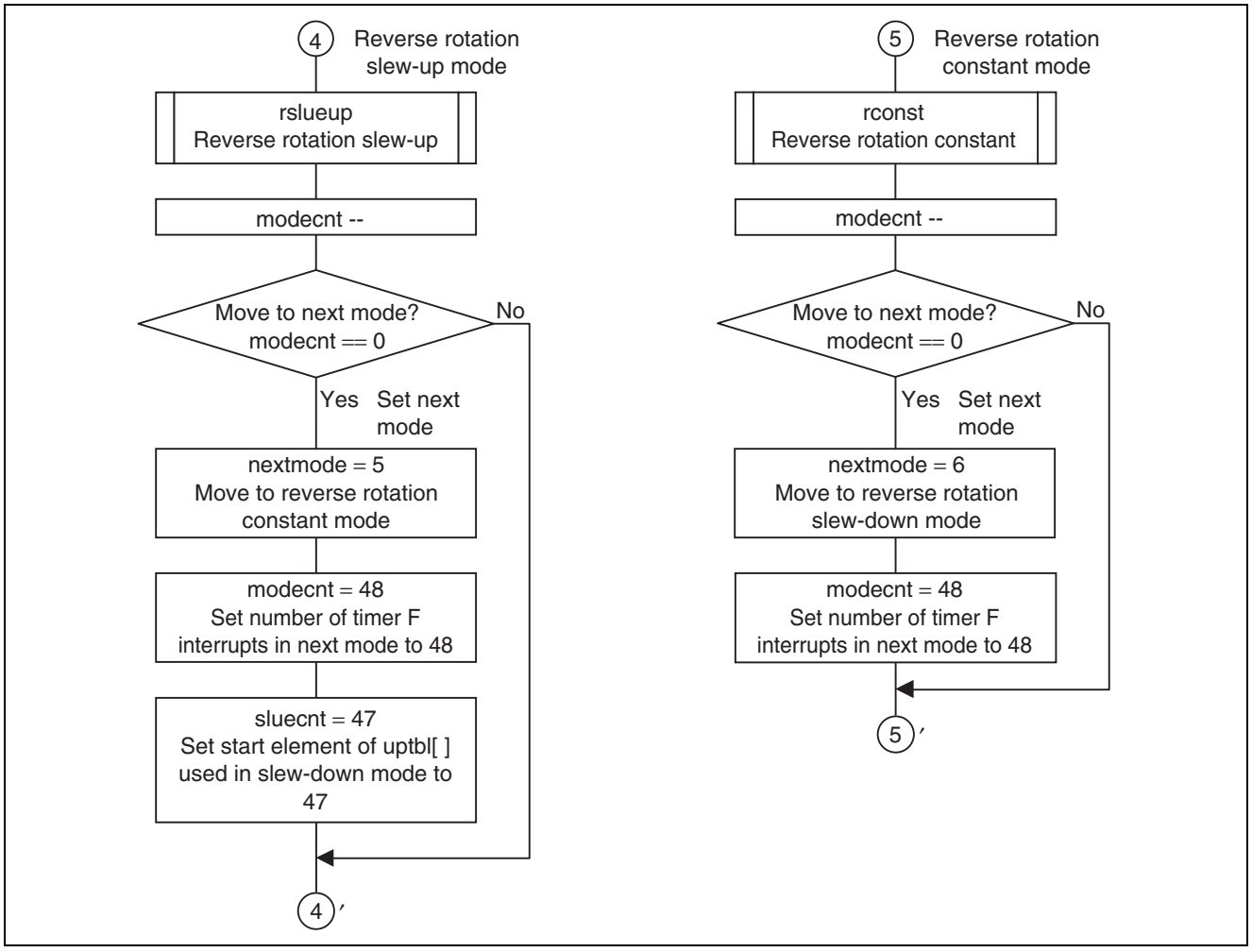
- IRR2 Interrupt request register 2 Address: H′FFF7

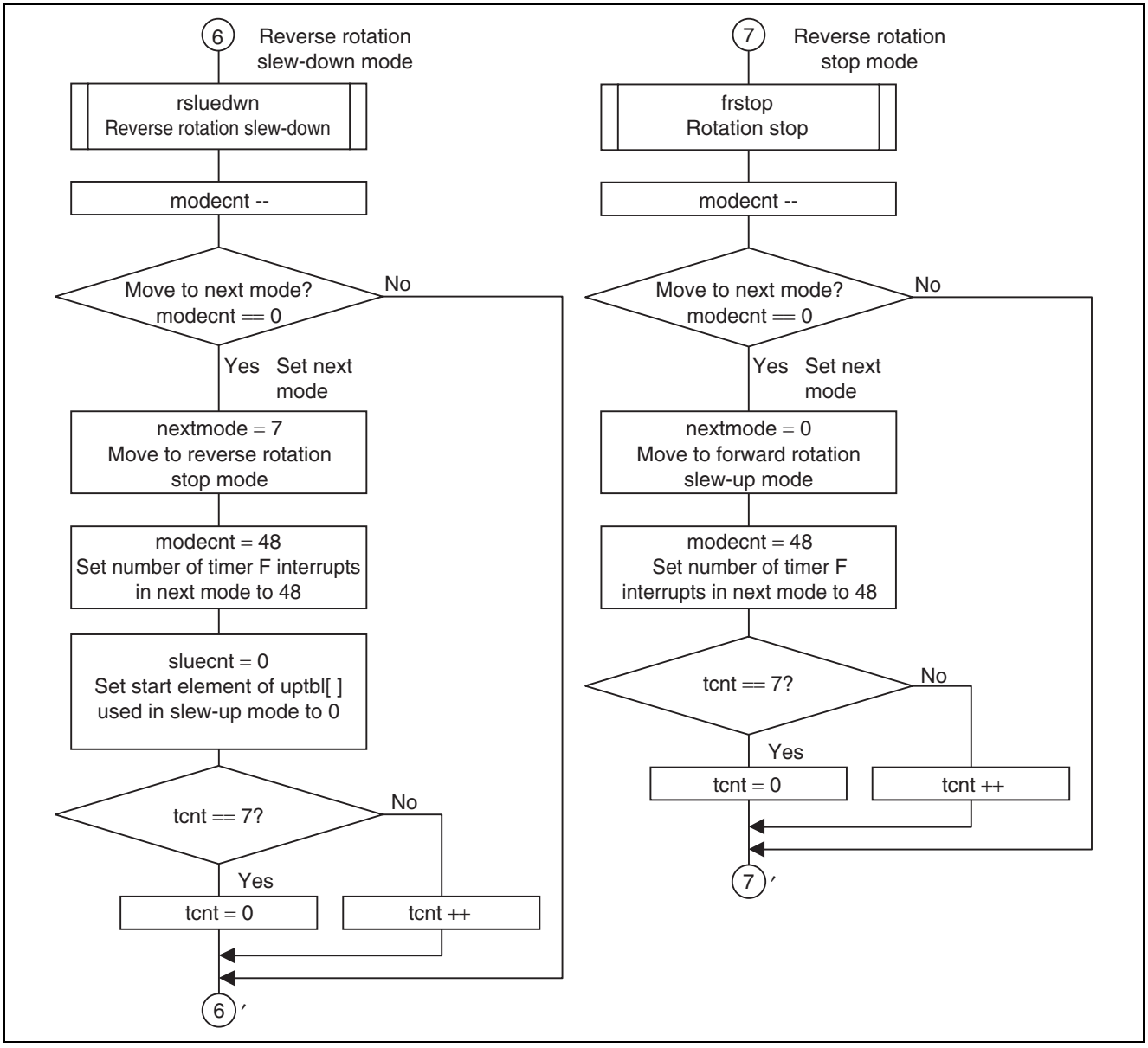
Bit	Bit Name	Setting	Function
3	IRRTFH	0	Timer FH interrupt request flag IRRTFH = 0: No timer FH interrupt request has been made. IRRTFH = 1: A timer FH interrupt request has been made.

3. Flowchart


tfhint
nextmode : ?
= 0
= 1
= 2
= 3
= 4
= 5
= 6
= 7
To forward rotation slew-up mode
To forward rotation constant mode
To forward rotation slew-down mode
To forward rotation stop mode
To reverse rotation slew-up mode
To reverse rotation constant mode
To reverse rotation slew-down mode
To reverse rotation stop mode
IRRTFH = 0
Clear timer FH interrupt flag
tmp = TCSRF
TCSRF = H'10
Clear flag
END
Forward rotation slew-up mode
fslueup
Forward rotation slew-up
modecnt --
Move to next mode?
modecnt == 0
No
Yes
Set next mode
nextmode = 1
Move to forward rotation constant mode
modecnt = 48
Set number of timer F interrupts in next mode to 48
sluecnt = 47
Set start element of uptbl[ ] used in slew-up mode to 47
Forward rotation constant mode
fconst
Forward rotation constant
modecnt --
Move to next mode?
modecnt == 0
No
Yes
Set next mode
nextmode = 2
Move to forward rotation slew-down mode
modecnt = 48
Set number of timer F interrupts in next mode to 48


2
Forward rotation
slew-down mode
fsluedwn
Forward rotation slew-down
modecnt --
Move to next mode?
modecnt == 0
No
Yes
Set next
mode
nextmode = 3
Move to forward rotation
stop mode
modecnt = 48
Set number of timer F
interrupts in next mode to 48
sluecnt = 0
Set start element of uptbl[ ]
used in slew-up mode to 0
tcnt == 0?
No
Yes
tcnt = 7
tcnt --
2'
3
Forward rotation
stop mode
frstop
Rotation stop
modecnt --
Move to next mode?
modecnt == 0
No
Yes
Set next
mode
nextmode = 4
Move to reverse rotation
slew-up mode
modecnt = 48
Set number of timer F
interrupts in next mode to 48
tcnt == 0?
No
Yes
tcnt = 7
tcnt --
3'


4
Reverse rotation slew-up mode
rslueup
Reverse rotation slew-up
modecnt --
Move to next mode?
modecnt == 0
No
Yes
Set next mode
nextmode = 5
Move to reverse rotation constant mode
modecnt = 48
Set number of timer F interrupts in next mode to 48
sluecnt = 47
Set start element of uptbl[ ] used in slew-down mode to 47
4'
5
Reverse rotation constant mode
rconst
Reverse rotation constant
modecnt --
Move to next mode?
modecnt == 0
No
Yes
Set next mode
nextmode = 6
Move to reverse rotation slew-down mode
modecnt = 48
Set number of timer F interrupts in next mode to 48
5'


6
Reverse rotation slew-down mode
rsluedwn
Reverse rotation slew-down
modecnt --
Move to next mode?
modecnt == 0
No
Yes
Set next mode
nextmode = 7
Move to reverse rotation stop mode
modecnt = 48
Set number of timer F interrupts in next mode to 48
sluecnt = 0
Set start element of uptbl[ ] used in slew-up mode to 0
tcnt == 7?
No
Yes
tcnt = 0
tcnt ++
6'
7
Reverse rotation stop mode
frstop
Rotation stop
modecnt --
Move to next mode?
modecnt == 0
No
Yes
Set next mode
nextmode = 0
Move to forward rotation slew-up mode
modecnt = 48
Set number of timer F interrupts in next mode to 48
tcnt == 7?
No
Yes
tcnt = 0
tcnt ++
7'

4.3.3 fslueup

1. Module specifications
   Function overview: Applies slew-up control during forward rotation.

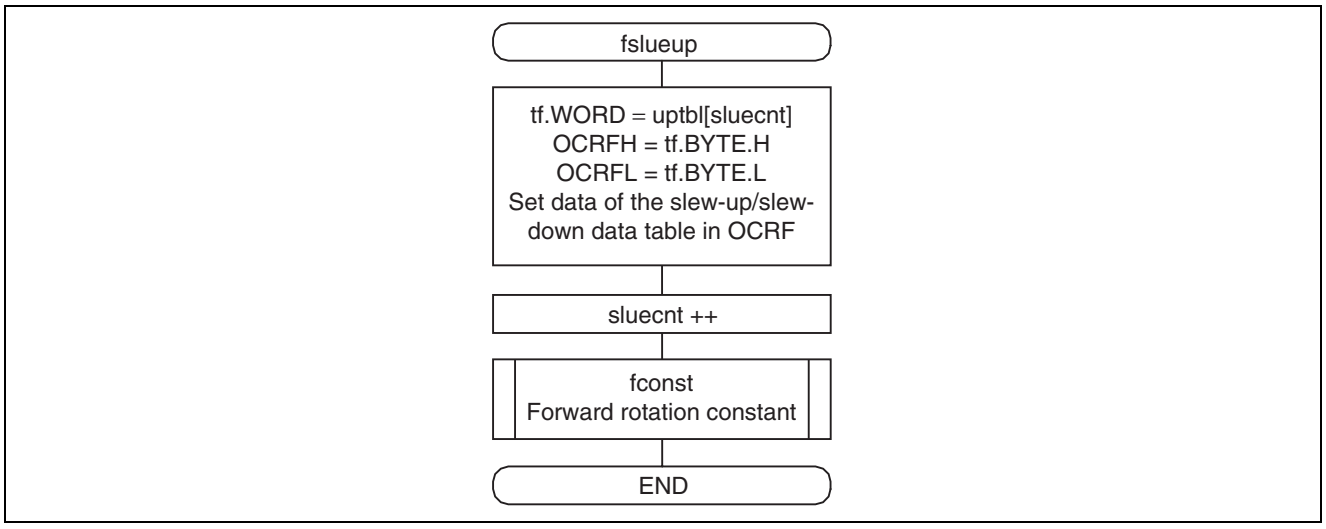
Table 6 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	sluecnt	Elements of array uptbl[] used for the slew-up and slew-down operation
ROM	unsigned short	uptbl[48]	Interrupt time data table for the slew-up and slew-down operation

2. Internal registers
   The internal registers used for this sample task are described below.

- OCRF 16-bit output compare register F Address: H′FFBA
  Function: A compare match occurs when a match is found between the setting of OCRF and the counter value of TCF.
  Setting: uptbl[sluecnt]

3. Flowchart



4.3.4 fsluedwn

1. Module specifications
   Function overview: Applies slew-down control during forward rotation.

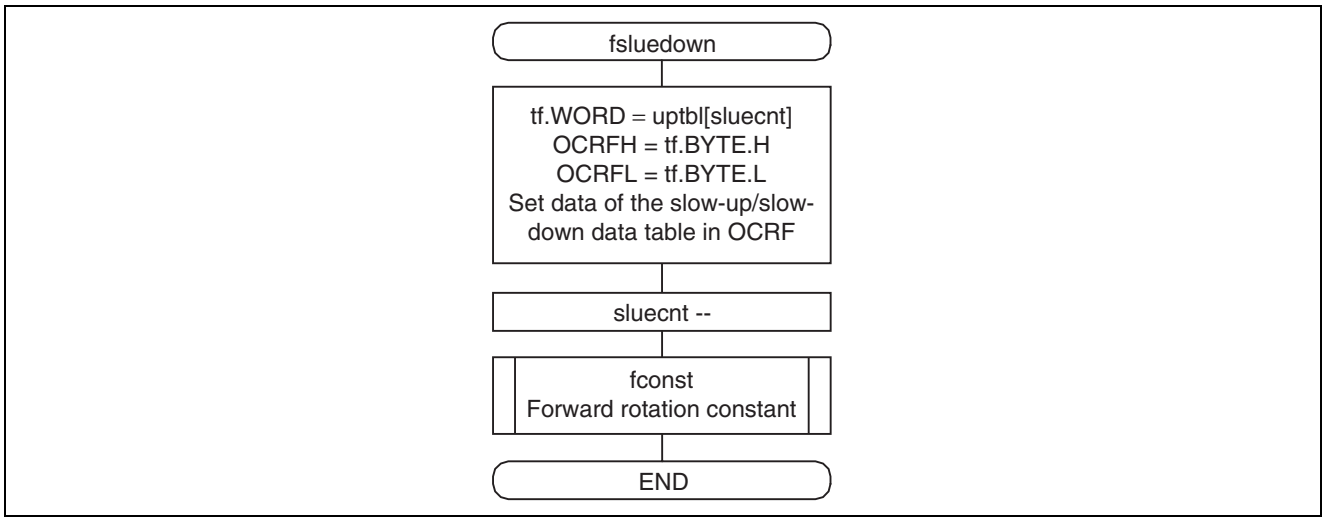
Table 7 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	sluecnt	Elements of array uptbl[] used for the slew-up and slew-down operation
ROM	unsigned short	uptbl[48]	Interrupt time data table for the slew-up and slew-down operation

2. Internal registers
   The internal registers used for this sample task are described below.

- OCRF 16-bit output compare register F Address: H′FFBA
  Function: A compare match occurs when a match is found between the setting of OCRF and the counter value of TCF.
  Setting: uptbl[sluecnt]

3. Flowchart



4.3.5 fconst

1. Module specifications
   Function overview: Applies constant control during forward rotation.

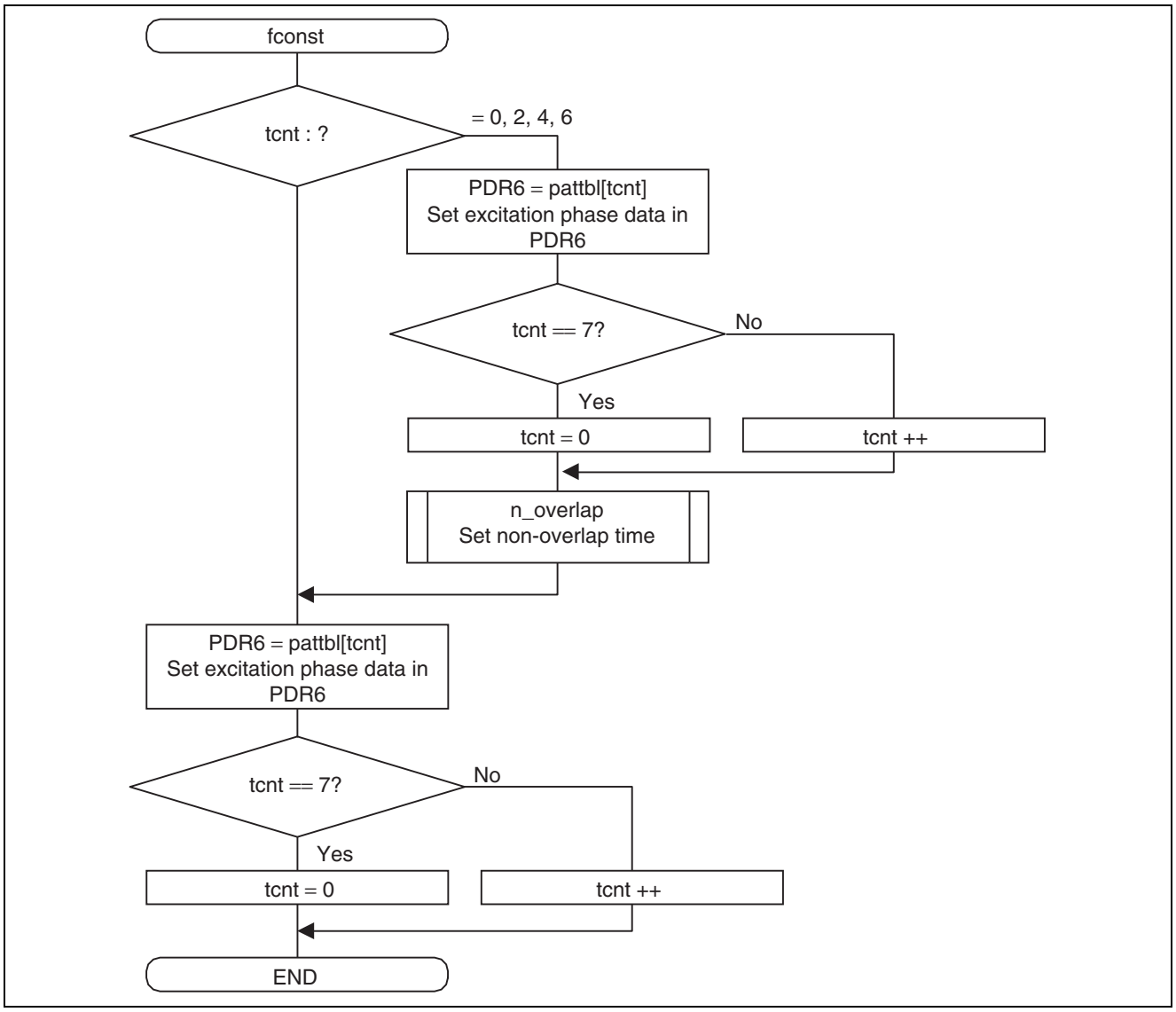
Table 8 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	tcnt	Elements of array pattbl[] representing stepper motor excitation data
ROM	unsigned char	pattbl[8]	Excitation pattern data table for the stepper motor

2. Internal registers
   The internal registers used for this sample task are described below.

- PDR6 Port data register 6 Address: H′FFD9
  Function: Uses P63 to P60 for excitation phase driving of the stepper motor.
  Setting: pattbl[tcnt]

3. Flowchart


fconst
tcnt : ?
= 0, 2, 4, 6
PDR6 = pattbl[tcnt]
Set excitation phase data in PDR6
tcnt == 7?
No
Yes
tcnt = 0
tcnt ++
n_overlap
Set non-overlap time
PDR6 = pattbl[tcnt]
Set excitation phase data in PDR6
tcnt == 7?
No
Yes
tcnt = 0
tcnt ++
END

4.3.6 frstop

1. Module specifications
   Function overview: Stops forward/reverse rotation.

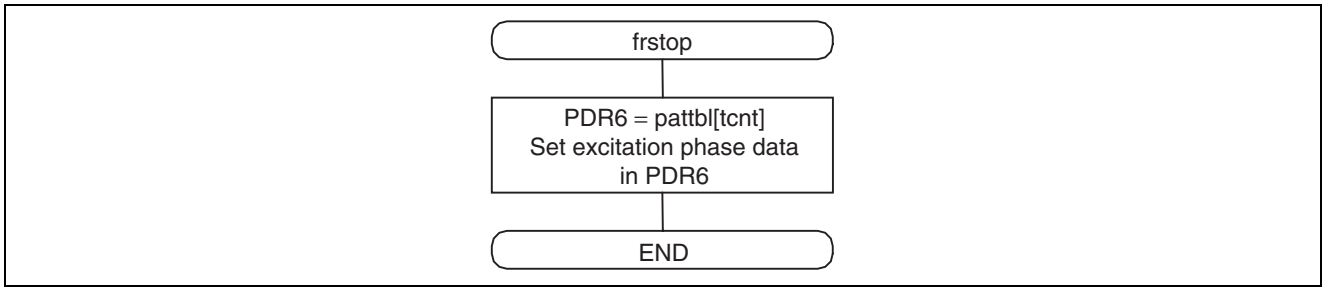
Table 9 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	tcnt	Elements of array pattbl[] representing stepper motor excitation data
ROM	unsigned char	pattbl[8]	Excitation pattern data table for the stepper motor

2. Internal registers
   The internal registers used for this sample task are described below.

- PDR6 Port data register 6 Address: H′FFD9
  Function: Uses P63 to P60 for excitation phase driving of the stepper motor.
  Setting: pattbl[tcnt]

3. Flowchart



4.3.7 rslueup

1. Module specifications
   Function overview: Applies slew-up control during reverse rotation.

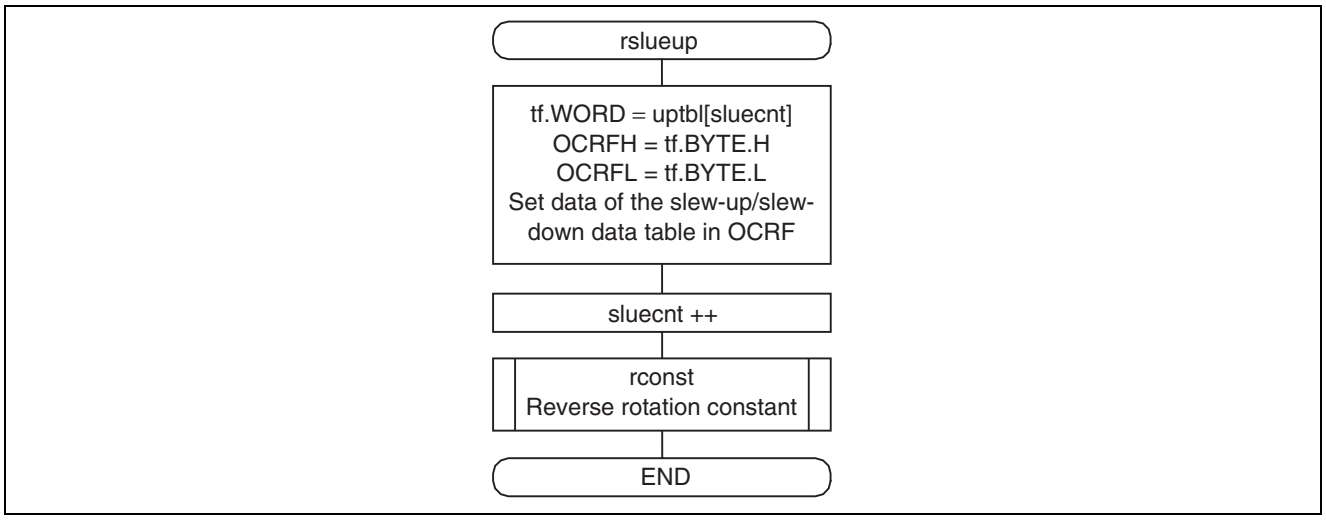
Table 10 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	sluecnt	Elements of array uptbl[] used for the slew-up and slew-down operation
ROM	unsigned short	uptbl[48]	Interrupt time data table for the slew-up and slew-down operation

2. Internal registers
   The internal registers used for this sample task are described below.

- OCRF 16-bit output compare register F Address: H′FFBA
  Function: A compare match occurs when a match is found between the setting of OCRF and the counter value of TCF.
  Setting: uptbl[sluecnt]

3. Flowchart



4.3.8 rsluedwn

1. Module specifications
   Function overview: Applies slew-down control during reverse rotation.

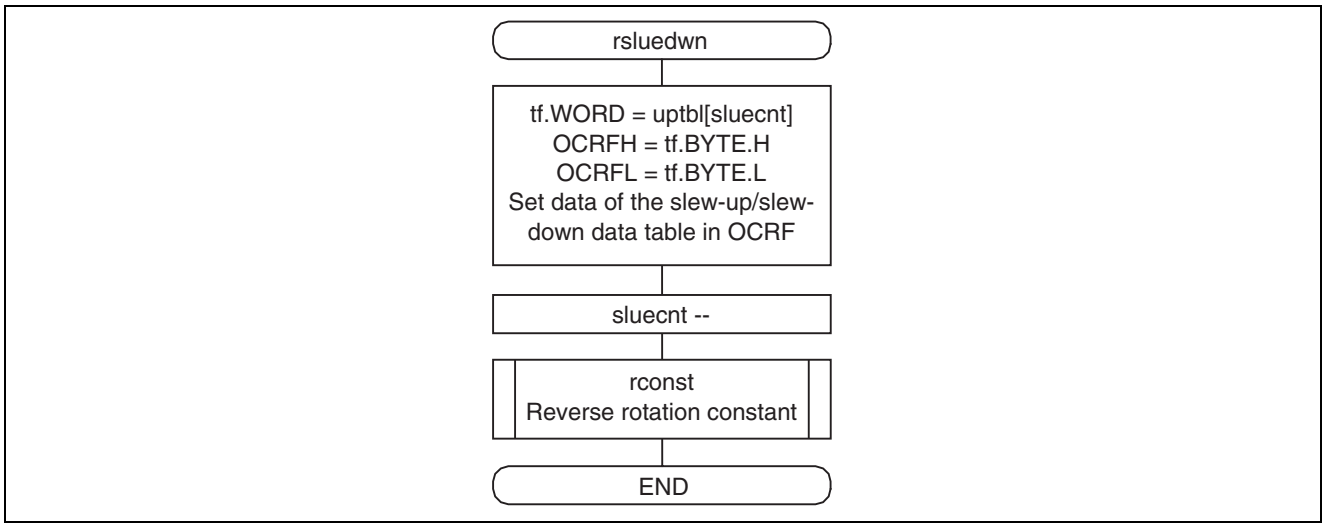
Table 11 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	sluecnt	Elements of array uptbl[] used for the slew-up and slew-down operation
ROM	unsigned short	uptbl[48]	Interrupt time data table for the slew-up and slew-down operation

2. Internal registers
   The internal registers used for this sample task are described below.

- OCRF 16-bit output compare register F Address: H′FFBA
  Function: A compare match occurs when a match is found between the setting of OCRF and the counter value of TCF.
  Setting: uptbl[sluecnt]

3. Flowchart



4.3.9 rconst

1. Module specifications
   Function overview: Applies constant control during reverse rotation.

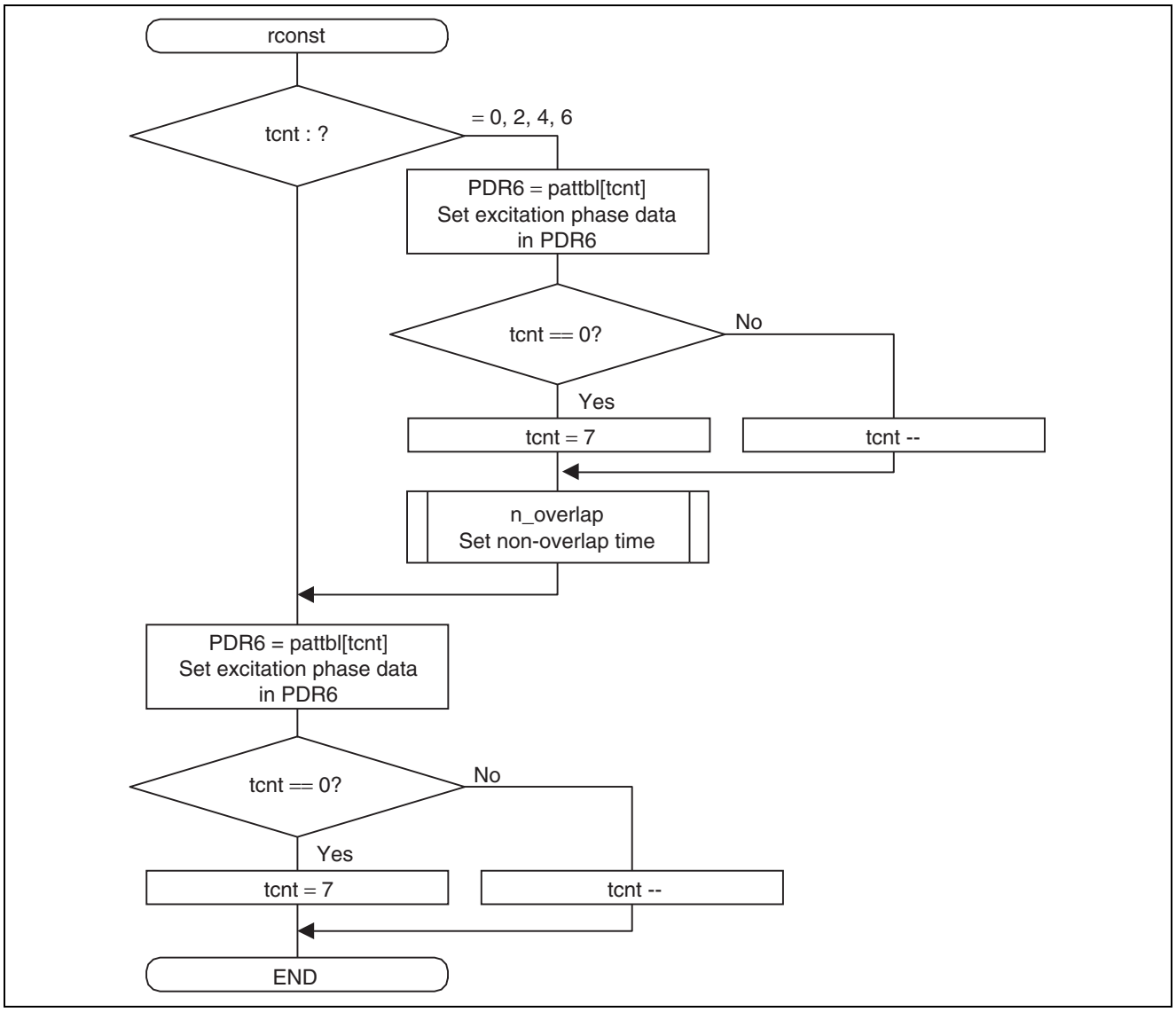
Table 12 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None
RAM	unsigned char	tcnt	Elements of array pattbl[] representing stepper motor excitation data
ROM	unsigned char	pattbl[8]	Excitation pattern data table for the stepper motor

2. Internal registers
   The internal registers used for this sample task are described below.

- PDR6 Port data register 6 Address: H′FFD9
  Function: Uses P63 to P60 for excitation phase driving of the stepper motor.
  Setting: pattbl[tcnt]

3. Flowchart


rconst
tcnt : ?
= 0, 2, 4, 6
PDR6 = pattbl[tcnt]
Set excitation phase data
in PDR6
tcnt == 0?
No
Yes
tcnt = 7
tcnt --
n_overlap
Set non-overlap time
PDR6 = pattbl[tcnt]
Set excitation phase data
in PDR6
tcnt == 0?
No
Yes
tcnt = 7
tcnt --
END

4.3.10 n_overlap

1. Module specifications
   Function overview: Sets the non-overlap time.

Table 13 Module Specifications

	Type	Variable Name	Description
Argument	None	None	None

2. Internal registers
   The internal registers used for this sample task are described below.

- IRR1 Interrupt request register 1 Address: H′FFF6

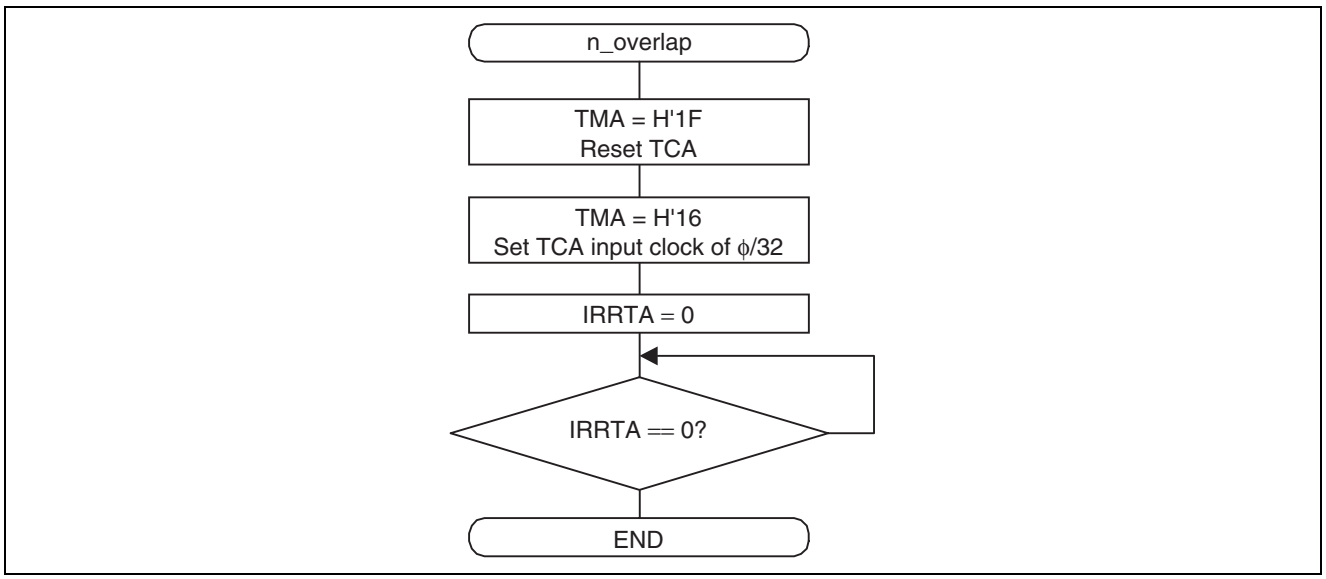
Bit	Bit Name	Setting	Function
3	IRRTA	0	Timer A interrupt request flag IRRTA = 0: No timer A interrupt request has been made. IRRTA = 1: A timer A interrupt request has been made.

- TMA Timer mode register A Address: H′FFB0

Bit	Bit Name	Setting	Function
3 to 0	TMA3 to TMA0	H′F	Internal clock select TMA3 to TMA0 = H′C, H′D, H′E, H′F: Resets TCA. TMA3 to TMA0 = H′6: Sets timer A as an interval function. Sets PSS as a TCA input clock source, and sets an overflow cycle of ϕ/32.

- TCA Timer counter A Address: H′FFB1
  Function: 8-bit up-counter to which ϕ/32 is input
  Setting: -

3. Flowchart



4.4 Link Address Specifications

Section Name	Address
CV1	H′0000
CV2	H′001E
P	H′0100
C	H′0800
DOUTDT	H′0810
B	H′FB80

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jul.16.04	—	First edition issued

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.