

Hitachi Single-Chip Microcomputer
H8/538 (On-chip I/O)
Application Note

Notice

When using this document, keep the following in mind:

1. This document may, wholly or partially, be subject to change without notice.
2. All rights are reserved: No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without Hitachi's permission.
3. Hitachi will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user's unit according to this document.
4. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.
5. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi, Ltd.
6. **MEDICAL APPLICATIONS:** Hitachi's products are not authorized for use in **MEDICAL APPLICATIONS** without the written consent of the appropriate officer of Hitachi's sales company. Such use includes, but is not limited to, use in life support systems. Buyers of Hitachi's products are requested to notify the relevant Hitachi sales offices when planning to use the products in **MEDICAL APPLICATIONS**.

Preface

The H8/538 is an original Hitachi high-performance, single-chip microcomputer built around a high-speed H8/500 CPU that includes on-chip peripheral functions, optimized for use as a controller in medium-scale machinery for industrial applications and office automation.

The chip contains a CPU, ROM, RAM, timers, and a serial communications interface that allow it to be used in a wide range of applications, from small systems to large. These application notes provide examples of tasks that use the peripheral functions of the H8/538. They are intended as a reference for user software design.

The task examples given in these application notes have been proven to work, but be sure to verify their functionality for yourself should you choose to use them in your applications.

Contents

Section 1	Guide to H8/538 Application Notes	1
1.1	Organization of the Information Provided in This Document	1
Section 2	H8/538 On-Chip I/Os	3
2.1	Pulse Output.....	3
2.1.1	Specifications.....	3
2.1.2	Functions Used	3
2.1.3	Operation	5
2.1.4	Software.....	6
2.1.5	Flowcharts.....	8
2.1.6	Program Lists.....	9
2.2	Measuring Pulse High Width and Low Width	13
2.2.1	Specifications.....	13
2.2.2	Functions Used	13
2.2.3	Operation	15
2.2.4	Software.....	16
2.2.5	Flowcharts.....	18
2.2.6	Program Lists.....	21
2.3	PWM Output 1.....	26
2.3.1	Specifications.....	26
2.3.2	Functions Used	27
2.3.3	Operation	29
2.3.4	Software.....	31
2.3.5	Flowcharts.....	33
2.3.6	Program Lists.....	34
2.4	Timer Counter Synchronized Preset and Simultaneous Count Start/Stop.....	39
2.4.1	Specifications.....	39
2.4.2	Functions Used	40
2.4.3	Operation	41
2.4.4	Software.....	42
2.4.5	Flowcharts.....	44
2.4.6	Program Lists.....	45
2.5	5-Phase Pulse Output Upon External Trigger	53
2.5.1	Specifications.....	53
2.5.2	Functions Used	54
2.5.3	Operation	56
2.5.4	Software.....	58
2.5.5	Flowcharts.....	60
2.5.6	Program Lists.....	61

2.6	PWM Output 2.....	65
2.6.1	Specifications.....	65
2.6.2	Functions Used	66
2.6.3	Operation	68
2.6.4	Software	70
2.6.5	Flowcharts.....	72
2.6.6	Program Lists.....	73
2.7	2-Phase Encoder Count.....	78
2.7.1	Specifications.....	78
2.7.2	Functions Used	78
2.7.3	Operation	79
2.7.4	Software	82
2.7.5	Flowcharts.....	84
2.7.6	Program Lists.....	85
2.8	2-Phase Encoder Count and 7-Phase PWM Output	89
2.8.1	Specifications.....	89
2.8.2	Functions Used	90
2.8.3	Operation	94
2.8.4	Software	98
2.8.5	Flowcharts.....	101
2.8.6	Program Lists.....	103
2.9	A/D Converter Startup by IPU and A/D Conversion	110
2.9.1	Specifications.....	110
2.9.2	Functions Used	110
2.9.3	Operation	112
2.9.4	Software	114
2.9.5	Flowcharts.....	116
2.9.6	Program Lists.....	118
2.10	Multiprocessor Communication	123
2.10.1	Specifications.....	123
2.10.2	Functions Used	123
2.10.3	Operation	126
2.10.4	Software	129
2.10.5	Flowcharts.....	132
2.10.6	Program Lists.....	135

Figures

Figure 1.1	Organization of the On-Chip I/O Application Notes	1
Figure 2.1	Pulse Output.....	3
Figure 2.2	Block Diagram of Pulse Output.....	4
Figure 2.3	Principles of Pulse Output Operation	5
Figure 2.4	Pulse Output Flowchart.....	8
Figure 2.5	Measuring the High Width and Low Width of the Pulse.....	13
Figure 2.6	Pulse Width Measurement Using Ch5 of the ITU	14
Figure 2.7	Pulse High Width and Low Width Measurement	15
Figure 2.8	Measuring Pulse Width (1 of 3) Flowchart.....	18
Figure 2.9	Measuring Pulse Width (2 of 3) Flowchart.....	19
Figure 2.10	Measuring Pulse Width (3 of 3) Flowchart.....	20
Figure 2.11	PWM Output Used to Generate the Sawtooth Waveform	26
Figure 2.12	PWM Output Block Diagram	27
Figure 2.13	Synchronized Clear Block Diagram	28
Figure 2.14	PWM Output (9-Phase) Used for Generating the Sawtooth Waveform.....	30
Figure 2.15	PWM Output 1 Flowchart.....	33
Figure 2.16	Timer Counter Synchronized Preset and Simultaneous Start/Stop.....	39
Figure 2.17	Block Diagram of Counter Synchronized Preset and Simultaneous Start/Stop	40
Figure 2.18	Operation of Counter Synchronization Preset and Simultaneous Start/Stop	41
Figure 2.19	Timer Counter Synchronized Preset and Simultaneous Count Start/Stop Flowchart	44
Figure 2.20	5-Phase Pulse Output from External Trigger.....	53
Figure 2.21	PWM Output Block Diagram	54
Figure 2.22	Block Diagram for Synchronized Clear Upon External Signal Rise	55
Figure 2.23	Operation for 5-Phase Pulse Output from External Trigger	57
Figure 2.24	5-Phase Pulse Output Flowchart.....	60
Figure 2.25	9-Phase PWM Output	65
Figure 2.26	PWM Output Block Diagram	66
Figure 2.27	Synchronized Clear Block Diagram	67
Figure 2.28	PWM Waveform (9-Phase) Used to Generate an Equilateral Triangle Waveform	69
Figure 2.29	PWM Output 2 Flowchart.....	72
Figure 2.30	2-Phase Encoder Count.....	78
Figure 2.31	Block Diagram of 2-Phase Encoder Count.....	79
Figure 2.32	Operation of 2-Phase Encoder Count (Upcount)	80
Figure 2.33	Operation of 2-Phase Encoder Count (Downcount)	81
Figure 2.34	2-Phase Encoder Count Flowchart.....	84
Figure 2.35	PWM Output (7-Phase) and 2-Phase Encoder Count	89
Figure 2.36	Block Diagram of 2-Phase Encoder Count.....	90
Figure 2.37	PWM Output Block Diagram	91
Figure 2.38	Synchronized Clear Block Diagram	92

Figure 2.39	PWM Waveform (7-Phase) Used to Generate an Equilateral Triangle Waveform	95
Figure 2.40	Operation of 2-Phase Encoder Count (Upcount).....	96
Figure 2.41	Operation of 2-Phase Encoder Count (Downcount).....	97
Figure 2.42	2-Phase Encoder Count and 7-Phase PWM Output (1 of 2).....	101
Figure 2.43	2-Phase Encoder Count and 7-Phase PWM Output (2 of 2).....	102
Figure 2.44	Block Diagram of Voltage Measurement Using the H8/538.....	110
Figure 2.45	Block Diagram of A/D Converter Startup and A/D Conversion	111
Figure 2.46	Principles of A/D Converter Operation	113
Figure 2.47	A/D Conversion (1 of 2) Flowchart.....	116
Figure 2.48	A/D Conversion (2 of 2) Flowchart.....	117
Figure 2.49	Asynchronous SCI Interface That Uses Multiprocessor Functions	123
Figure 2.50	Master SCI Block Diagram.....	124
Figure 2.51	Slave SCI Block Diagram.....	125
Figure 2.52	Multiprocessor Communication (Master Side).....	126
Figure 2.53	Slave Side of SCI (Operation when the ID of Another Station is Received)	127
Figure 2.54	Block Diagram of Slave Side of SCI (Operation when Own ID is Received)	128
Figure 2.55	Multiprocessor Communication Operation (Master Side).....	132
Figure 2.56	Multiprocessor Communication Operation (Slave Side).....	133
Figure 2.57	Multiprocessor Communication Flowchart	134

Tables

Table 2.1	IPU Function Allocation	4
Table 2.2	Modules.....	6
Table 2.3	Arguments.....	6
Table 2.4	Internal Registers Used.....	6
Table 2.5	General Registers Used.....	7
Table 2.6	IPU (Ch5) Function Allocation.....	14
Table 2.7	Modules.....	16
Table 2.8	Arguments.....	16
Table 2.9	Internal Registers Used.....	17
Table 2.10	General Registers Used.....	17
Table 2.11	IPU Function Allocation	29
Table 2.12	Modules.....	31
Table 2.13	Arguments.....	31
Table 2.14	Internal Registers Used.....	32
Table 2.15	General Registers Used.....	32
Table 2.16	IPU Function Allocation	40
Table 2.17	Modules.....	42
Table 2.18	Internal Registers Used.....	42
Table 2.19	General Registers Used.....	43
Table 2.20	IPU Function Allocation	56
Table 2.21	Modules.....	58

Table 2.22	Arguments.....	58
Table 2.23	Internal Registers Used.....	59
Table 2.24	General Registers Used.....	59
Table 2.25	IPU Function Allocation.....	68
Table 2.26	Modules.....	70
Table 2.27	Arguments.....	70
Table 2.28	Internal Registers Used.....	71
Table 2.29	General Registers Used.....	71
Table 2.30	IPU Function Allocation.....	79
Table 2.31	Modules.....	82
Table 2.32	Arguments.....	82
Table 2.33	Internal Registers Used.....	82
Table 2.34	General Registers Used.....	83
Table 2.35	IPU Function Allocation.....	93
Table 2.36	Modules.....	98
Table 2.37	Arguments.....	98
Table 2.38	Internal Registers Used.....	99
Table 2.39	General Registers Used.....	100
Table 2.40	A/D Converter and IPU Function Allocation.....	112
Table 2.41	Modules.....	114
Table 2.42	Arguments.....	114
Table 2.43	Internal Registers Used.....	115
Table 2.44	General Registers Used.....	115
Table 2.45	RAM.....	115
Table 2.46	SCI Function Allocation.....	125
Table 2.47	Modules.....	129
Table 2.48	Arguments.....	129
Table 2.49	Internal Registers Used.....	129
Table 2.50	General Registers Used.....	129
Table 2.51	Modules.....	130
Table 2.52	Arguments.....	130
Table 2.53	Internal Registers Used.....	130
Table 2.54	General Registers Used.....	131
Table 2.55	RAM Used.....	131

Section 1 Guide to H8/538 Application Notes

The application notes describe the use of peripheral functions (timer, serial communications, etc.) using examples of simple tasks (such as pulse output).

1.1 Organization of the Information Provided in This Document

The explanation for each peripheral function is organized as shown in figure 1.1.

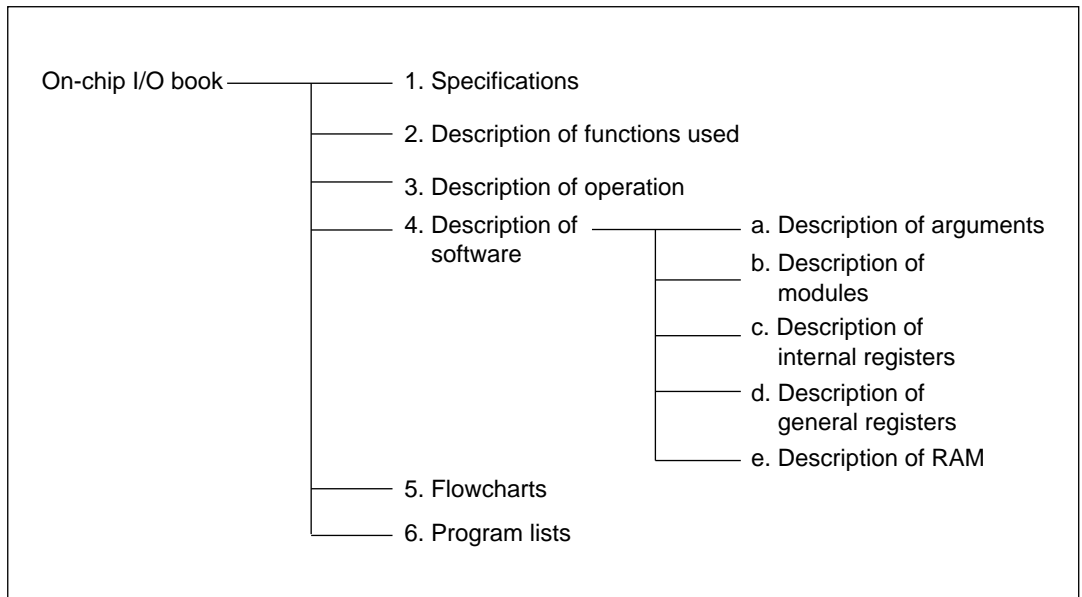


Figure 1.1 Organization of the On-Chip I/O Application Notes

1. Specifications

Describes the system specifications of the task examples.

2. Functions Used

Describes the features of the peripheral functions used in the task examples, and the allocation of peripheral functions.

3. Operation

Describes the operation of the task examples, using timing charts.

4. Software

- Modules: Describes the software modules used to make the task examples run.

- **Arguments:** Describes the input arguments needed to run the modules, and the output arguments.
- **Internal registers:** Describes the internal registers of the peripheral functions set by the modules (Timer Control register, Serial Mode register, etc.).
- **General registers:** Gives the names of (R0—R7) and describes the functions of the general registers used by the modules.
- **RAM:** Gives the label names of the RAM used by the modules and describes the functions.

5. Flowcharts

Describes the software that runs the task examples using general flowcharts.

6. Program Lists

Gives the program lists of the software that runs the task examples.

Section 2 H8/538 On-Chip I/Os

2.1 Pulse Output

Functions used: IPU (compare match toggle output)

2.1.1 Specifications

As shown in figure 2.1, a 50% duty pulse is output from the T4IOC1 pin of ch4. When operating at 10 MHz, the pulse cycle can be set between 16 μ s and 6.5 ms (in 200 ns units) and the number of pulses output can be set anywhere between 1 and 127.

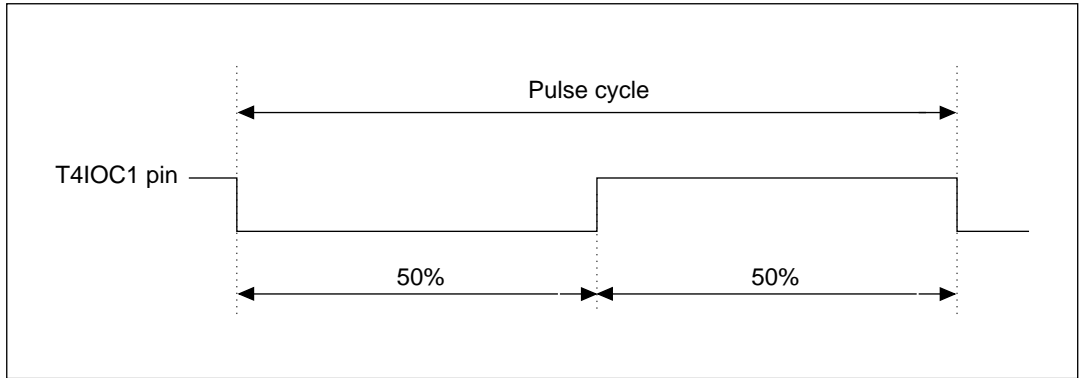


Figure 2.1 Pulse Output

2.1.2 Functions Used

This task example uses the following functions of the IPU:

- Function for outputting pulses automatically using hardware without software intervention (output-compare output)
- Function for inverting the output level every time a compare match occurs (toggle output)
- Function for clearing the timer counters upon GR1 compare match

In this task example, these functions are used to output pulses (figure 2.2).

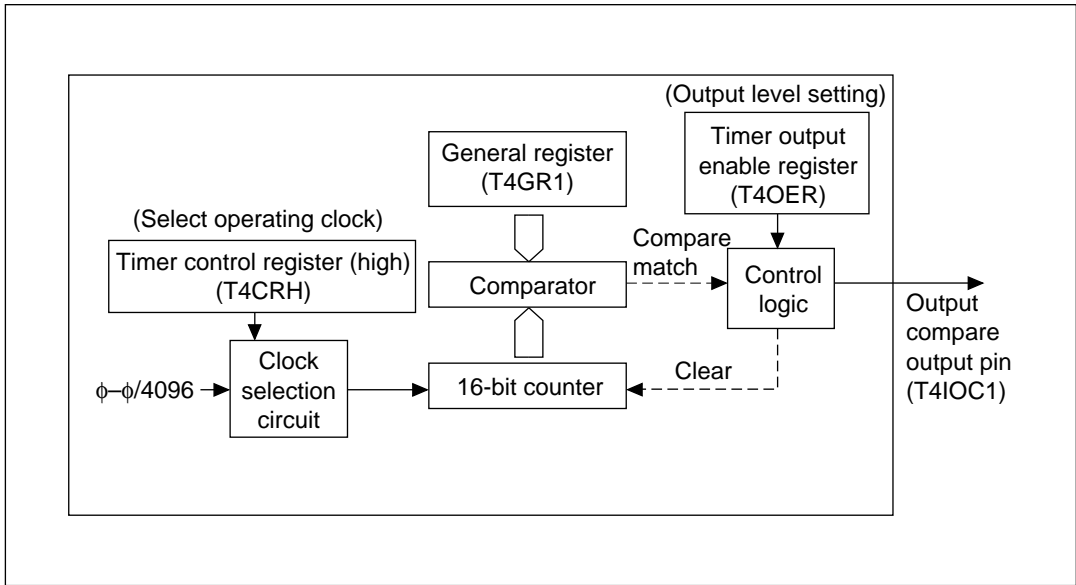


Figure 2.2 Block Diagram of Pulse Output

Table 2.1 lists the function allocations for this task example. IPU functions are allocated to output pulses, as listed in table 2.1.

Table 2.1 IPU Function Allocation

IPU Function	Function
T4IOC1	Outputs pulse.
T4CRL	Selects ch4 timer counter clear sources.
T4SRH	Selects ch4 compare match interrupt enable/disable.
T4OER	Selects ch4 compare match signal output enable/disable and output level.
T4GR1	Sets 1/2 the timer counter value of the output pulse cycle.

2.1.3 Operation

Figure 2.3 lists the principles of operation. Pulses are output by H8/538 hardware processing and software processing.

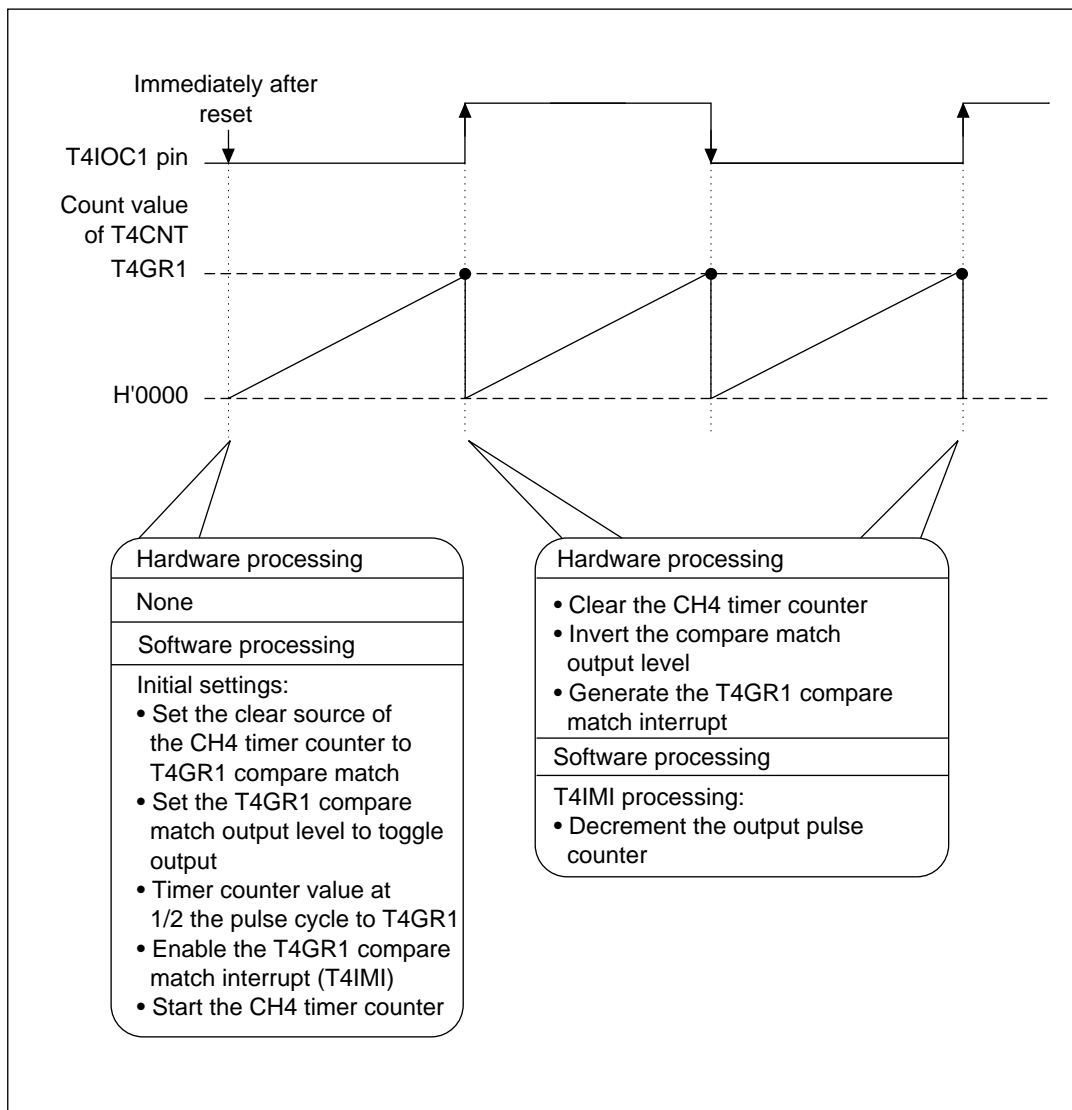


Figure 2.3 Principles of Pulse Output Operation

2.1.4 Software

Tables 2.2 through 2.5 list IPU function information.

Table 2.2 Modules

Module Name	Label Name	Function
Main routine	POUTMN	Initializes pulse output.
Compare match interrupt	POUTI	Halts the count of number of output pulses and pulse output.

Table 2.3 Arguments

Label Name	Function	Data Length	Name of Module Used	I/O
PLS_CYC	Sets the timer value corresponding to pulse cycle. The pulse cycle is found by the following equation. Pulse cycle (ns) = timer value \times system clock cycle (100 ns when operating at 10 MHz) \times ch4 input clock division ratio.	2 bytes	Main routine	I
PLS_CNT	Sets the number of pulses output. Sets a value twice the number of pulses actually output.	1 byte	Main routine Pulse output	O I

Table 2.4 Internal Registers Used

Register Name	Function	Name of Module Used
T4CRL	Selects ch4 timer counter clear sources.	Main routine
T4SRH	Selects enable/disable of ch4 compare match interrupt	Main routine Compare match interrupt
T4SRL	Indicates the ch4 status.	Main routine Compare match interrupt
T4OER	Selects enable/disable of the ch4 compare match signal output and output level.	Main routine Compare match interrupt
T4GR1	Sets 1/2 the timer counter value of the output pulse cycle.	Main routine

Table 2.5 General Registers Used

Name of Module Used	Name of Register	Function
Main routine	R0	Used as work space when setting data.
Pulse output	R0	Used as work space when setting data.

RAM:

This task example does not use RAM except for arguments.

2.1.5 Flowcharts

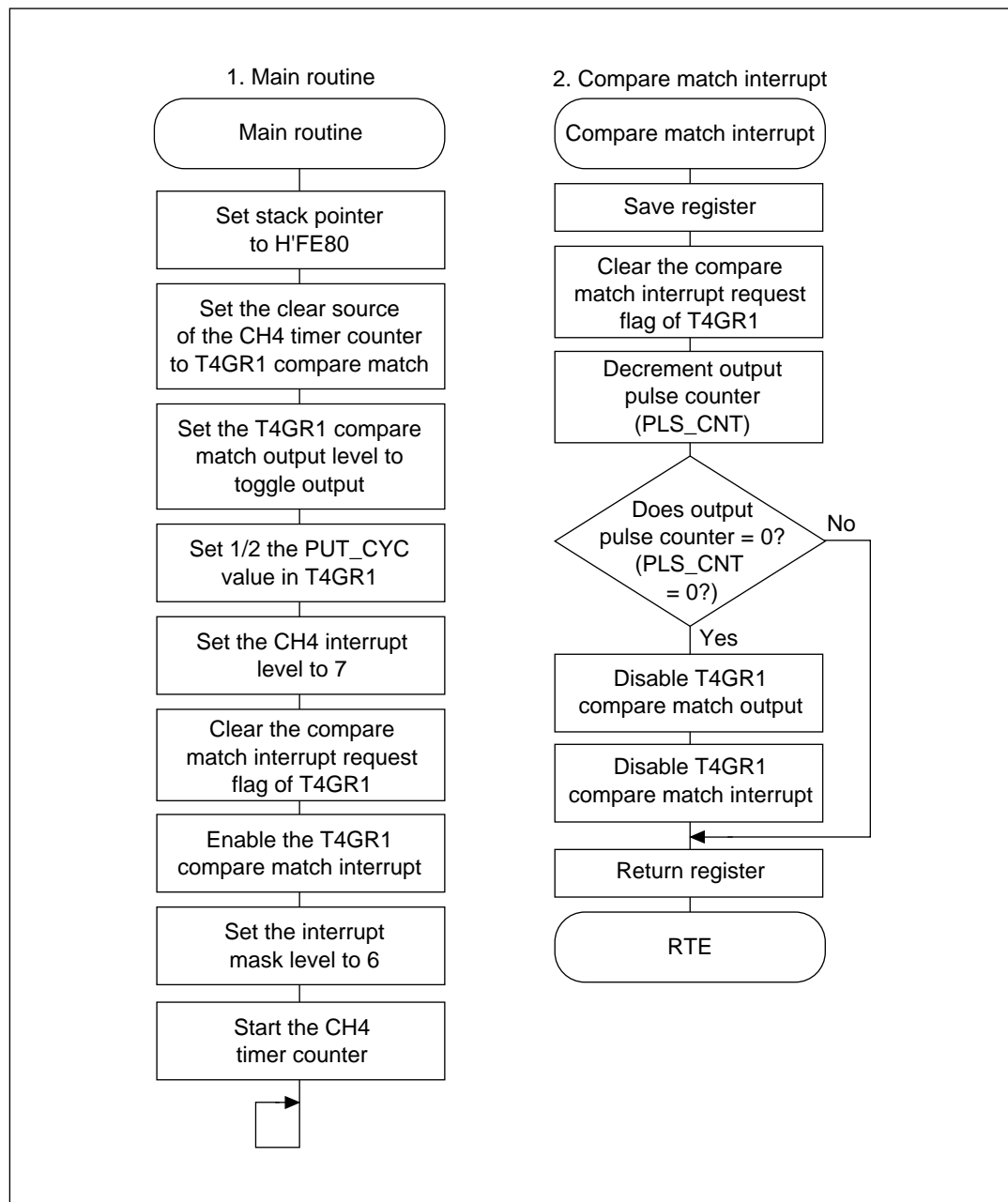


Figure 2.4 Pulse Output Flowchart

2.2 Measuring Pulse High Width and Low Width

Functions used: IPU (input capture)

2.2.1 Specifications

1. As shown in figure 2.5, the high width and low width of the pulse input to T5IOC1 are measured, and the results are stored in RAM.
2. The results of measurement are indicated by the timer count value as a 24-digit binary number (2^0 — 2^{23}).
3. When operating at 10 MHz, the high width and low width of the pulse can be measured between 18 μ s and 1.67 s (in 100 ns units).

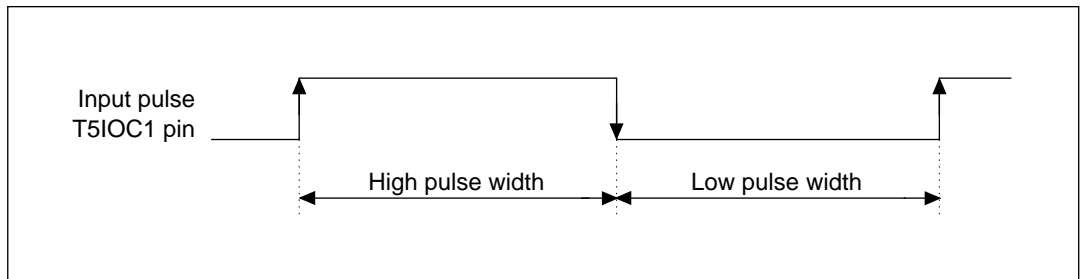


Figure 2.5 Measuring the High Width and Low Width of the Pulse

2.2.2 Functions Used

Figure 2.6 is a block diagram that lists ch5 of the IPU used by this task example. Ch5 has the following functions:

- Function for detecting input pulse rise and fall edges and setting the timer values at those times to internal registers (input capture)
- Function for detecting pulse rise and fall edges and starting up interrupt processing (input capture interrupt)
- Function for starting interrupt processing when a timer counter overflow occurs (overflow interrupt)
- Function for clearing the timer counter upon input capture

In this task example, these functions are used as shown in figure 2.6 to measure high and low width.

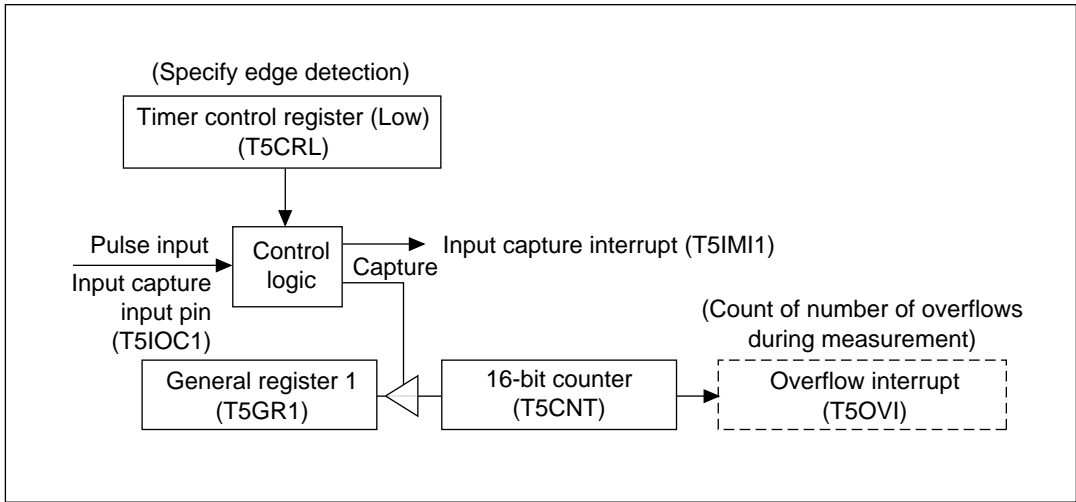


Figure 2.6 Pulse Width Measurement Using Ch5 of the ITU

Table 2.6 lists the function allocations for this task example. IPU ch5 functions are allocated to measure the high and low widths.

Table 2.6 IPU (Ch5) Function Allocation

IPU (Ch5)	Function
T5IOC1	Input pin for measurement pulse.
T5IMI1	Starts up the input capture interrupt upon detection of a rise or fall edge of the measurement pulse.
T5GR1	Sets the timer counter value at detection of a rise or fall edge of the measurement pulse.
T5OVI	Starts up the overflow interrupt upon overflow of the ch5 timer counter.

2.2.3 Operation

Figure 2.7 lists the principles of operation. Pulse high and low widths are measured by H8/538 hardware processing and software processing.

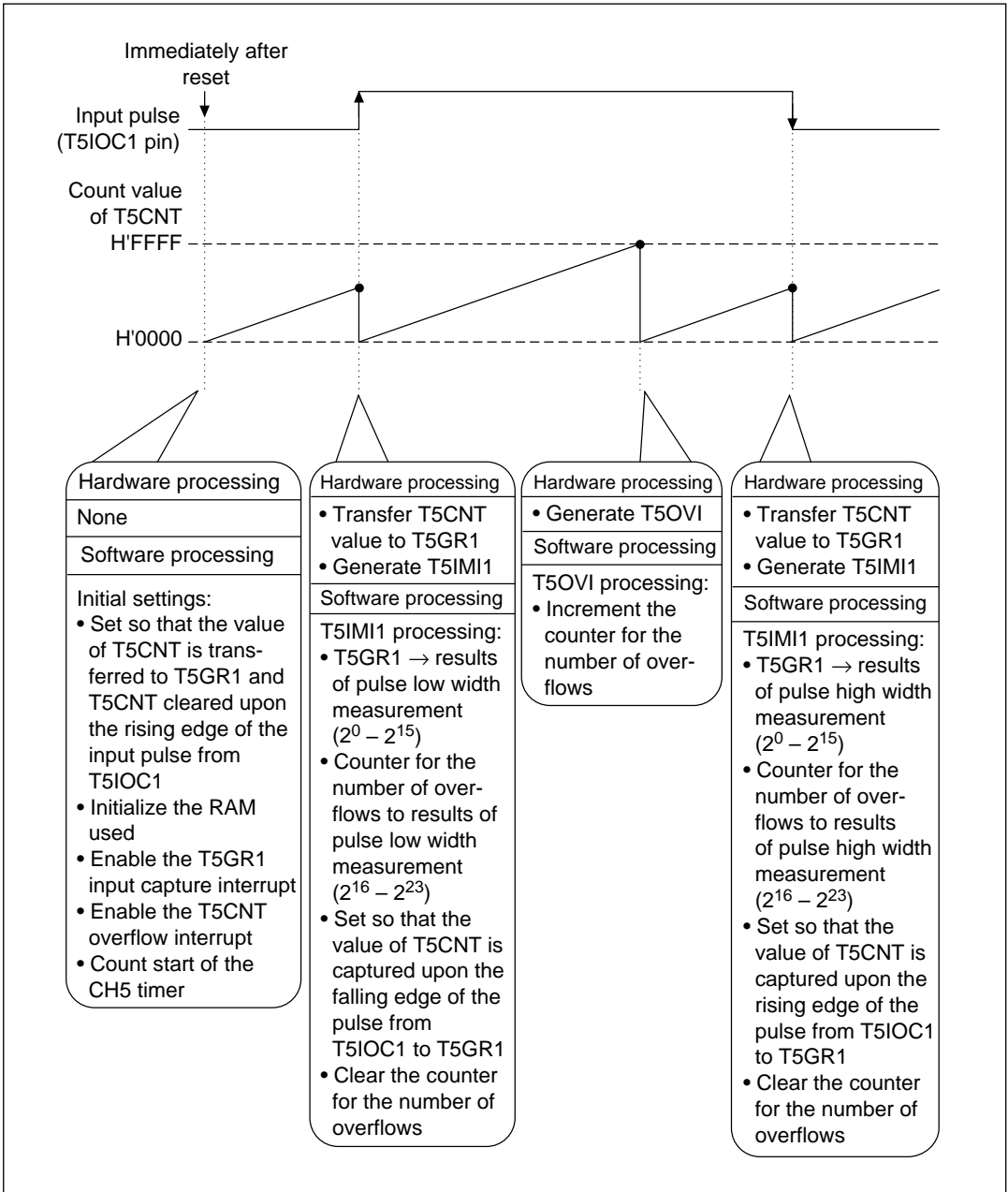


Figure 2.7 Pulse High Width and Low Width Measurement

2.2.4 Software

Tables 2.7 through 2.10 list information for this function.

Table 2.7 Modules

Module Name	Label Name	Function
Main routine	PWIDMN	Initializes IPU and RAM.
Input capture interrupt	EDGE	Starts up upon T5IMI and measures the High and Low widths of the pulse from the T5GR1 value and number of overflows that occur.
Overflow interrupt	OVFC	Starts up upon T5OVI and counts the number of overflows that occur.

Table 2.8 Arguments

Label Name, Register Name	Function	Data Length	Name of Module Used	I/O
OVF_TIMC	Indicates the number of overflows that occur.	1 byte	Overflow interrupt Input capture interrupt	O I
HWID_L16	Sets the results of measuring the High width of the pulse to bits 2 ⁰ —2 ¹⁵ .	2 bytes	Input capture interrupt	O
HWID_U8	Sets the results of measuring the High width of the pulse to bits 2 ¹⁶ —2 ²³ .	1 byte	Overflow interrupt Input capture interrupt	O I
LWID_L16	Sets the results of measuring the Low width of the pulse to bits 2 ⁰ —2 ¹⁵ .	2 bytes	Input capture interrupt	O
LWID_U8	Sets the results of measuring the Low width of the pulse to bits 2 ¹⁶ —2 ²³ .	1 byte	Overflow interrupt Input capture interrupt	O I
HWID_ERF	Set when the measurable pulse High width is exceeded.	1 bit	Overflow interrupt	O
LWID_ERF	Set when the measurable pulse Low width is exceeded.	1 bit	Overflow interrupt	O

Table 2.9 Internal Registers Used

Register Name	Function	Name of Module Used
T5SRH	Selects enable/disable of ch5 interrupts.	Main routine
T5SRL	Indicates the ch5 status.	Main routine Input capture interrupt Overflow interrupt
T5GR1	Sets the T5CNT value at the rise edge and fall edge of the pulse and measures the pulse High and Low widths with that value.	Main routine Input capture interrupt
T5CRL	Selects timer counter clear sources and input capture edge.	Overflow interrupt

Table 2.10 General Registers Used

Name of Module Used	Name of Register	Function
Input capture interrupt	R0	Used as work space when setting data.
Overflow interrupt	R0	Used as work space when setting data.

RAM:

This task example does not use RAM except for arguments.

2.2.5 Flowcharts

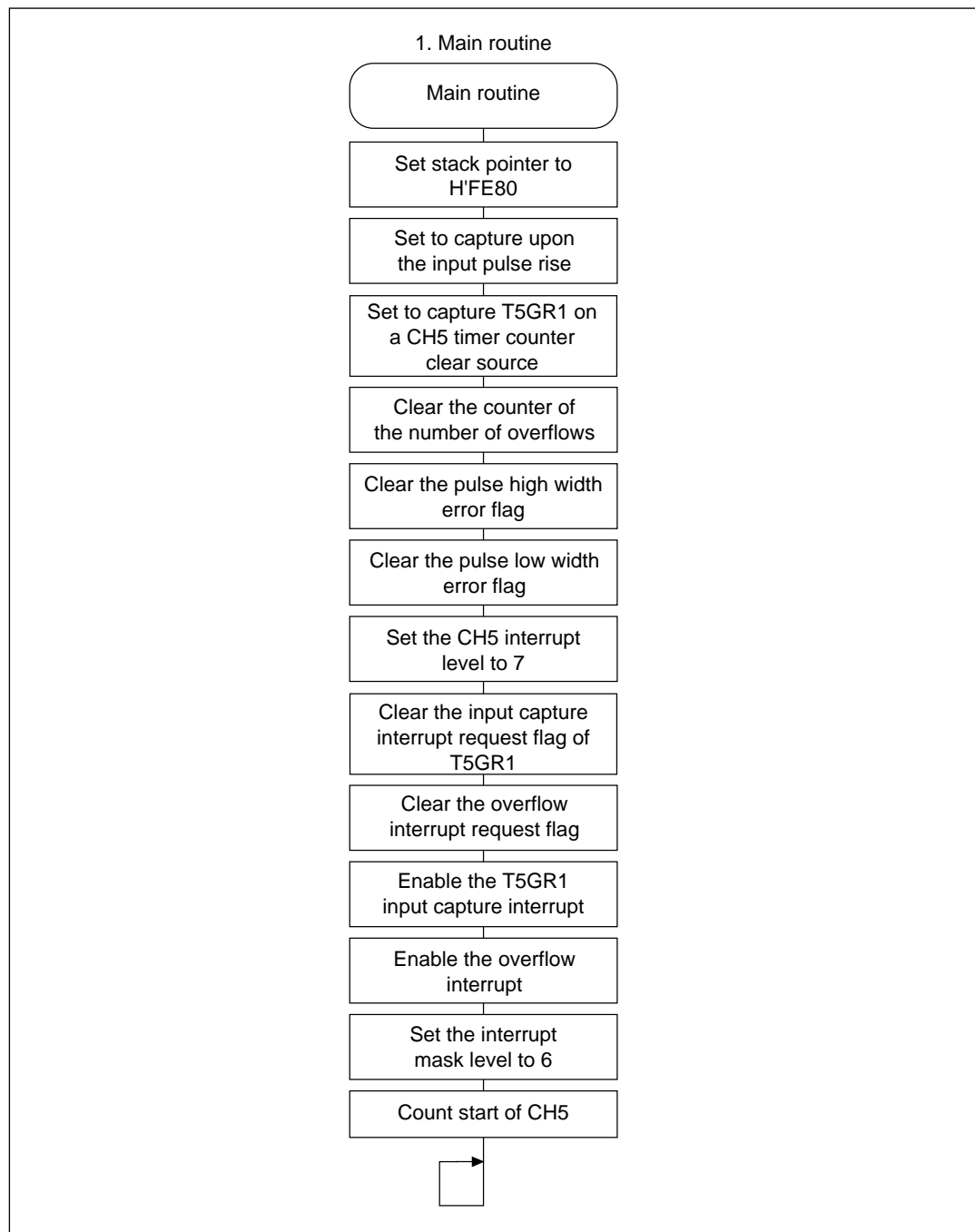


Figure 2.8 Measuring Pulse Width (1 of 3) Flowchart

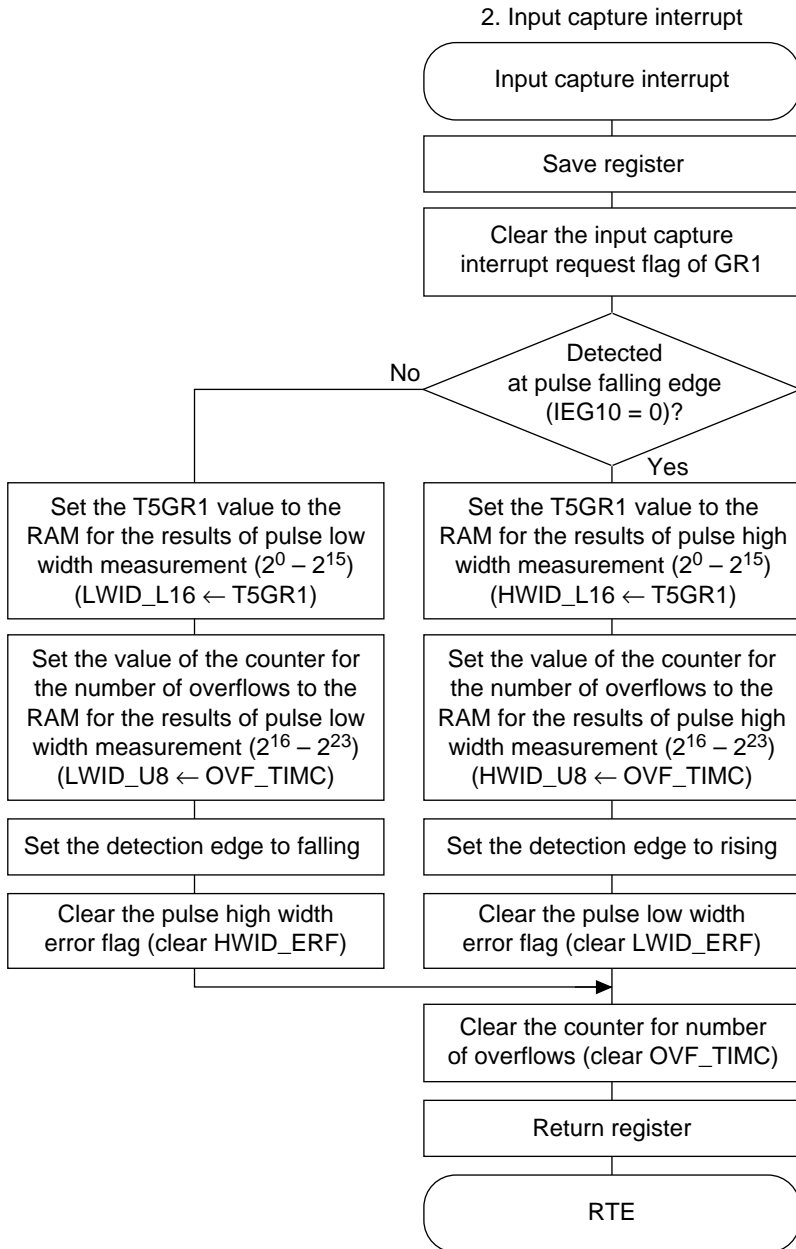


Figure 2.9 Measuring Pulse Width (2 of 3) Flowchart

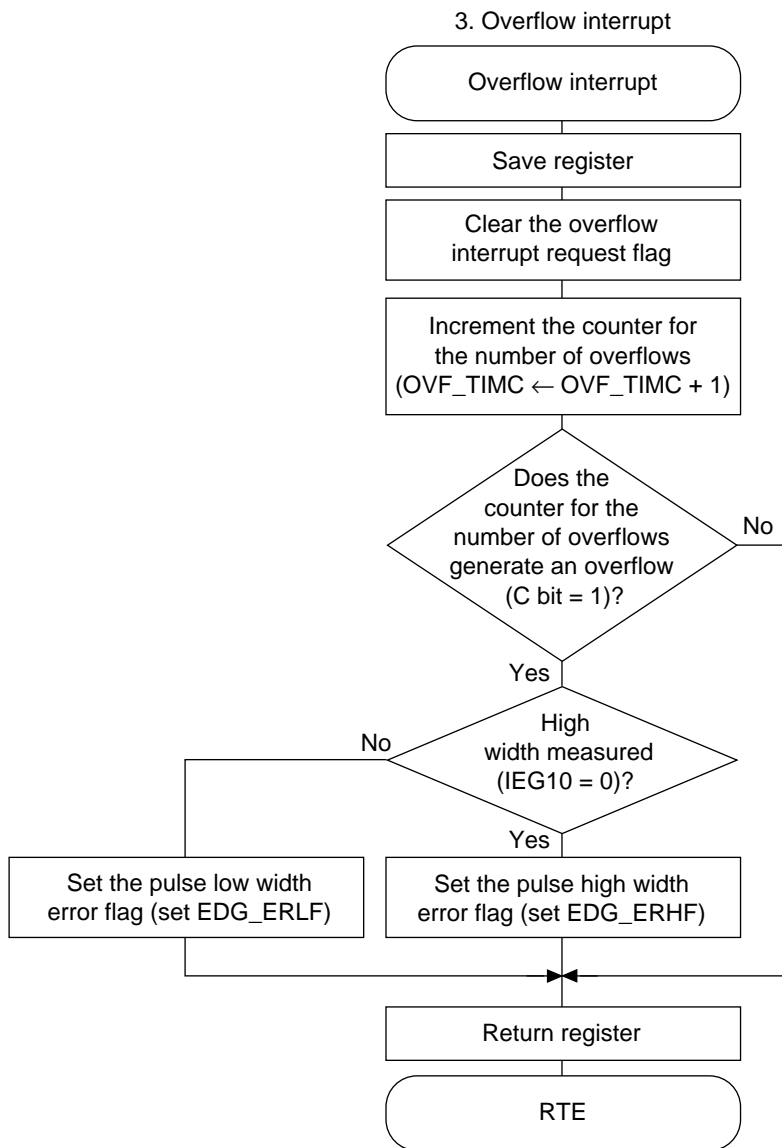


Figure 2.10 Measuring Pulse Width (3 of 3) Flowchart

2.3 PWM Output 1

Functions used: IPU

2.3.1 Specifications

As shown in figure 2.11, a PWM waveform (9-phase) is output for use in generating a sawtooth waveform. When operating at 10 MHz, the PWM cycle can be set anywhere between 1 μ s and 6.55 ms.

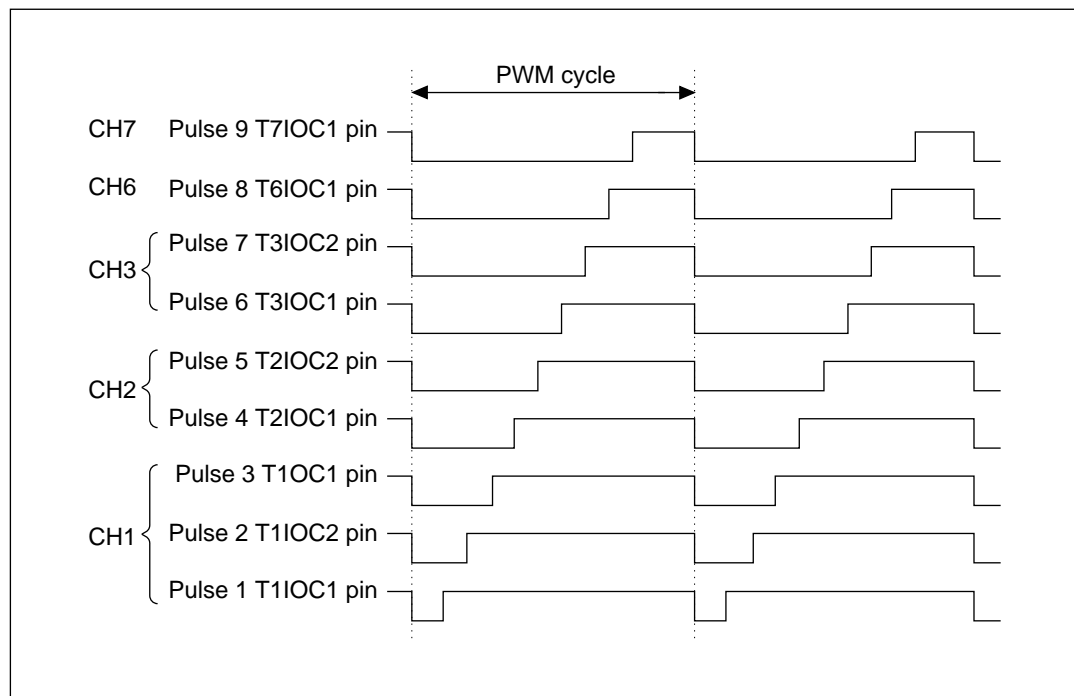


Figure 2.11 PWM Output Used to Generate the Sawtooth Waveform

This task example uses the following functions of the IPU.

- Figure 2.12 is a block diagram of PWM output (ch1 only); figure 2.13 is a block diagram of a synchronized clear.



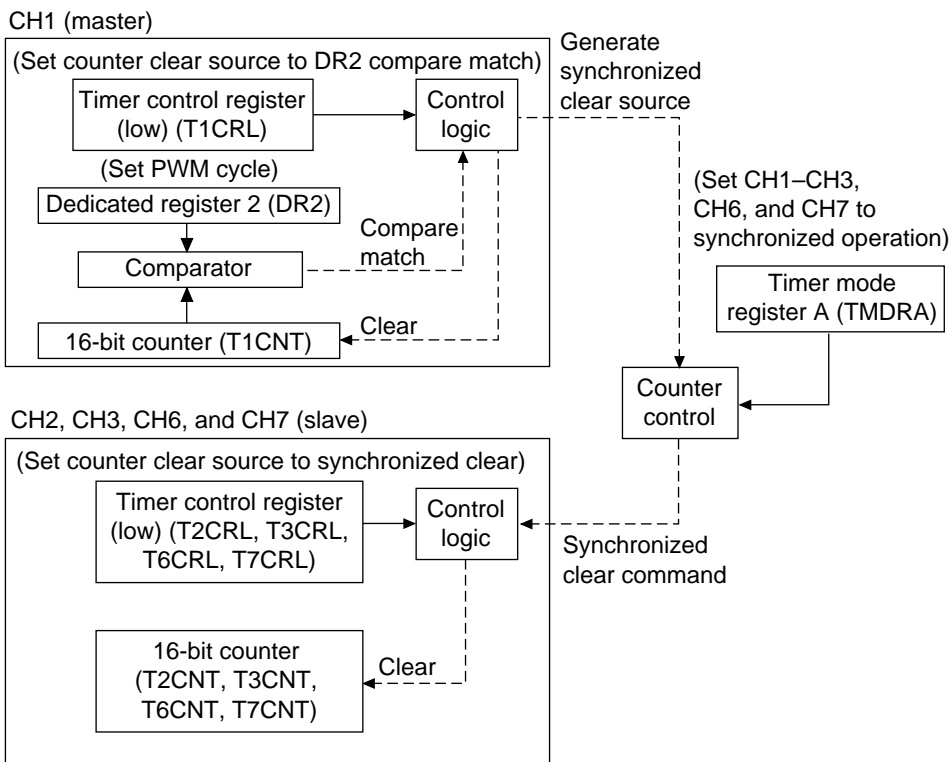


Figure 2.13 Synchronized Clear Block Diagram

Table 2.11 lists the function allocations for this task example. IPU functions are allocated as shown in table 2.11, and PWM pulses output.

Table 2.11 IPU Function Allocation

IPU Function	Function
T1IOC1—3, T2IOC1—2, T3IOC1—2, T6IOC1, T7IOC1	Pins for PWM pulse output.
T1CRL—T3CRL, T6CRL, T7CRL	Selects timer counter clear sources for ch1—ch3, ch6 and ch7.
TMDRA	Synchronizes operation of ch1—ch3, ch6 and ch7 timer counters.
TMDRB	Makes ch1—ch3, ch6 and ch7 operate as PWM timers.
T1DR2	Sets the PWM cycle.
T1GR1	Sets the timer counter value that makes T1IOC1 output high.
T1GR3	Sets the timer counter value that makes T1IOC1 output low.
T1GR2	Sets the timer counter value that makes T1IOC2 output high.
T1GR4	Sets the timer counter value that makes T1IOC2 output low.
T1DR1	Sets the timer counter value that makes T1IOC1 output high.
T1DR3	Sets the timer counter value that makes T1IOC1 output low.
T2GR1	Sets the timer counter value that makes T2IOC1 output high.
T2DR1	Sets the timer counter value that makes T2IOC1 output low.
T2GR2	Sets the timer counter value that makes T2IOC2 output high.
T2DR2	Sets the timer counter value that makes T2IOC2 output low.
T3GR1	Sets the timer counter value that makes T3IOC1 output high.
T3DR1	Sets the timer counter value that makes T3IOC1 output low.
T3GR2	Sets the timer counter value that makes T3IOC2 output high.
T3DR2	Sets the timer counter value that makes T3IOC2 output low.
T6GR1	Sets the timer counter value that makes T6IOC1 output high.
T6GR2	Sets the timer counter value that makes T6IOC1 output low.
T7GR1	Sets the timer counter value that makes T7IOC1 output high.
T7GR2	Sets the timer counter value that makes T7IOC1 output low.
TSTR	Starts ch1—ch3, ch6 and ch7 as timer counters.

2.3.3 Operation

Figure 2.14 shows the principles of operation. A 9-phase PWM waveform is output for use in generating a sawtooth waveform from the PWM output pins of ch1—ch7 (everything except T1IOC1, T2IOC1 and T3IOC1 is omitted) by H8/538 hardware processing and software processing.

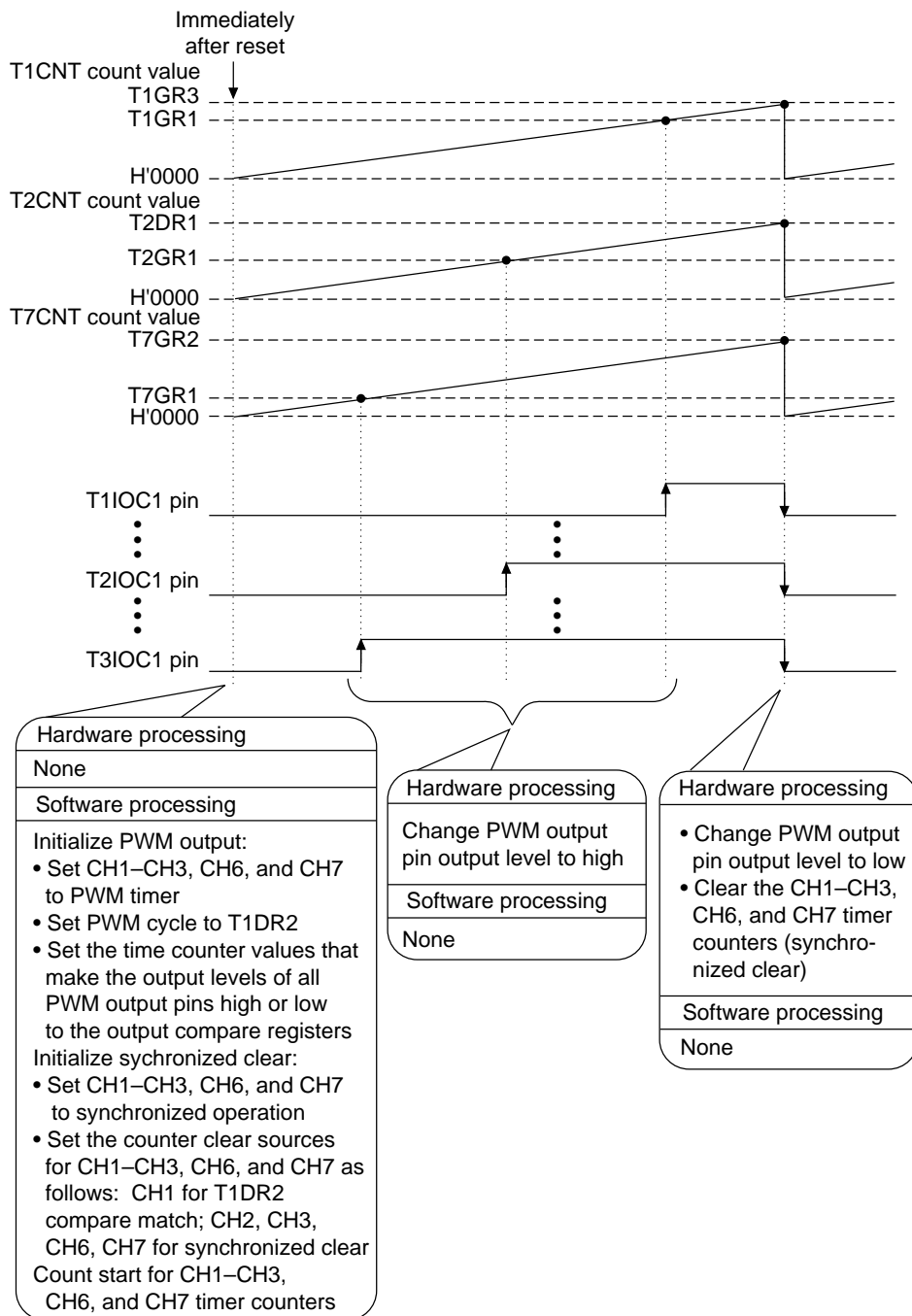


Figure 2.14 PWM Output (9-Phase) Used for Generating the Sawtooth Waveform

2.3.4 Software

Tables 2.12 through 2.15 list software information for this function.

Table 2.12 Modules

Module Name	Label Name	Function
Main routine	PWMOUT1	Sets for synchronized clear and PWM output for ch1—ch3, ch6 and ch7.

Table 2.13 Arguments

Label Name	Function	Data Length	Name of Module Used	I/O
PWM_HI1 — PWM_H19	Sets the timer counter value corresponding to High width of the pulse. The pulse high width is found by the following equation. Pulse high width (ns) = timer counter value × system clock cycle (100 ns when operating at 10 MHz) × input clock division ratio for each channel.	2 bytes	Main routine	I
PWM_CYC	Sets the timer value corresponding to PWM cycle. The PWM cycle is found by the following equation. PWM cycle (ns) = timer counter value × system clock cycle (100 ns when operating at 10 MHz) × input clock division ratio for each channel.	2 bytes	Main routine	I

Table 2.14 Internal Registers Used

Register Name	Function	Module Used
T1CRL—T3CRL, T6CRL, T7CRL	Selects timer counter clear sources.	Main routine
TMDRA	Selects channels to operate in sync.	
TMDRB	Makes channels operate as PWM timers.	
T1DR2	Sets the PWM cycle.	
T1GR1	Sets the timer counter value that makes T1IOC1 output high.	
T1GR3	Sets the timer counter value that makes T1IOC1 output low.	
T1GR2	Sets the timer counter value that makes T1IOC2 output high.	
T1GR4	Sets the timer counter value that makes T1IOC2 output low.	
T1DR1	Sets the timer counter value that makes T1IOC1 output high.	
T1DR3	Sets the timer counter value that makes T1IOC1 output low.	
T2GR1	Sets the timer counter value that makes T2IOC1 output high.	
T2DR1	Sets the timer counter value that makes T2IOC1 output low.	
T2GR2	Sets the timer counter value that makes T2IOC2 output high.	
T2DR2	Sets the timer counter value that makes T2IOC2 output low.	
T3GR1	Sets the timer counter value that makes T3IOC1 output high.	
T3DR1	Sets the timer counter value that makes T3IOC1 output low.	
T3GR2	Sets the timer counter value that makes T3IOC2 output high.	
T3DR2	Sets the timer counter value that makes T3IOC2 output low.	
T6GR1	Sets the timer counter value that makes T6IOC1 output high.	
T6GR2	Sets the timer counter value that makes T6IOC1 output low.	
T7GR1	Sets the timer counter value that makes T7IOC1 output high.	
T7GR2	Sets the timer counter value that makes T7IOC1 output low.	
TSTR	Starts and stops ch1—ch7 as timer counters.	

Table 2.15 General Registers Used

Name of Module Used	Name of Register	Function
Main routine	R0—R2	Used as work space when setting data.

RAM:

This task example does not use RAM except for arguments.

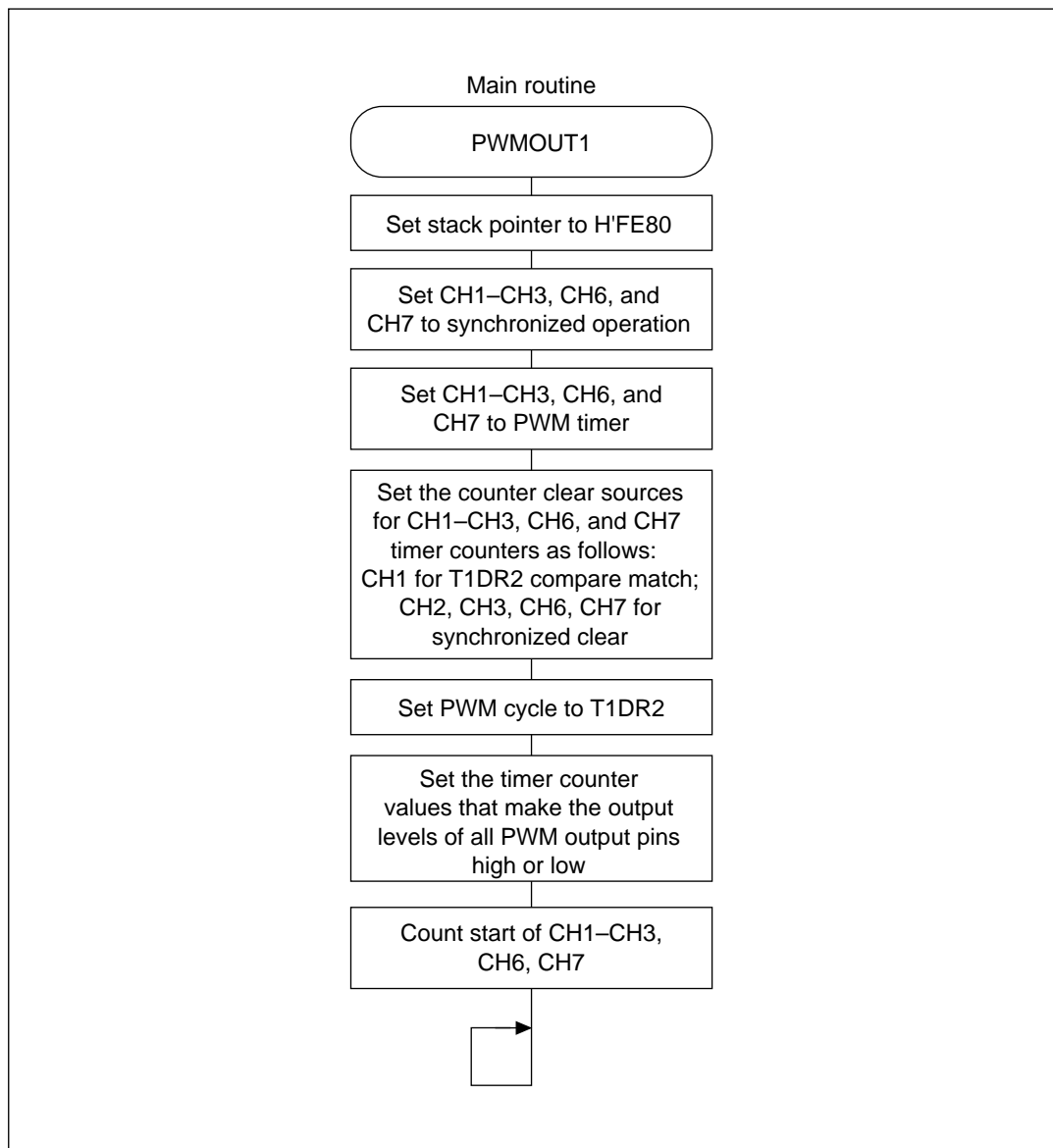


Figure 2.15 PWM Output 1 Flowchart

2.4 Timer Counter Synchronized Preset and Simultaneous Count Start/Stop

Functions used: IPU

2.4.1 Specifications

As shown in figure 2.16, seven 50% duty pulses are output from ch1—ch7. These pulses simultaneously stop the ch1—ch7 counters upon IRQ1 fall, preset the counters to H'0000, and start them upon IRQ2 fall.

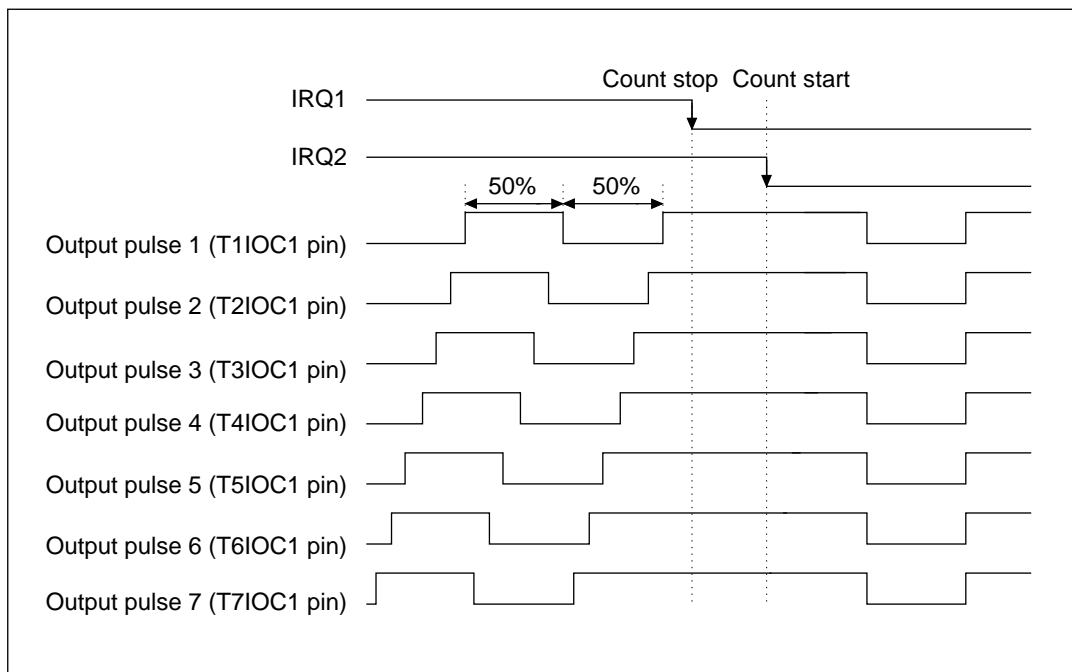


Figure 2.16 Timer Counter Synchronized Preset and Simultaneous Start/Stop

2.4.2 Functions Used

This task example uses the following functions of the IPU.

- Function for simultaneously starting or stopping multiple timer counters
- Function for simultaneously rewriting the values of multiple timer counters. Figure 2.17 is a block diagram of synchronized preset and simultaneous start/stop of counters (synchronized preset).

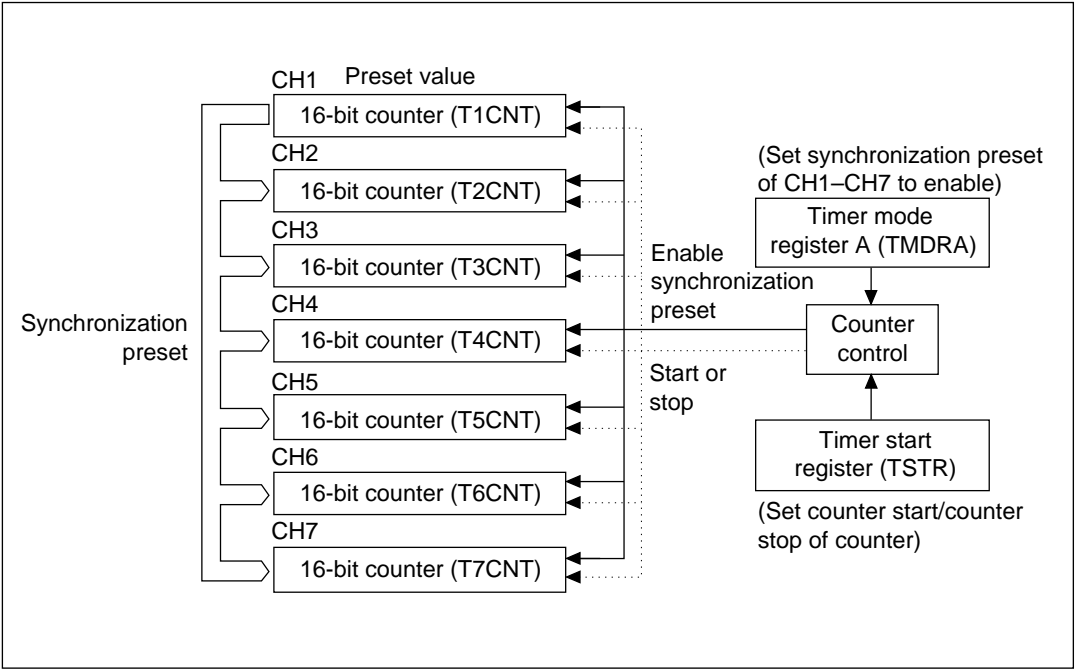


Figure 2.17 Block Diagram of Counter Synchronized Preset and Simultaneous Start/Stop

Table 2.16 lists the function allocations for this task example. IPU functions are allocated as listed in table 2.16, and synchronized preset and start/stop of the ch1—ch7 timer counters is performed.

Table 2.16 IPU Function Allocation

IPU Function	Function
TSTR	Starts or stops the ch1—ch7 timer counters.
TMDRA	Enables the ch1—ch7 timer counter synchronized preset.
T1CNT	Sets the preset value.

2.4.3 Operation

Figure 2.18 shows the principles of operation. Counters are preset in sync and simultaneously started/stopped using H8/538 hardware processing and software processing.

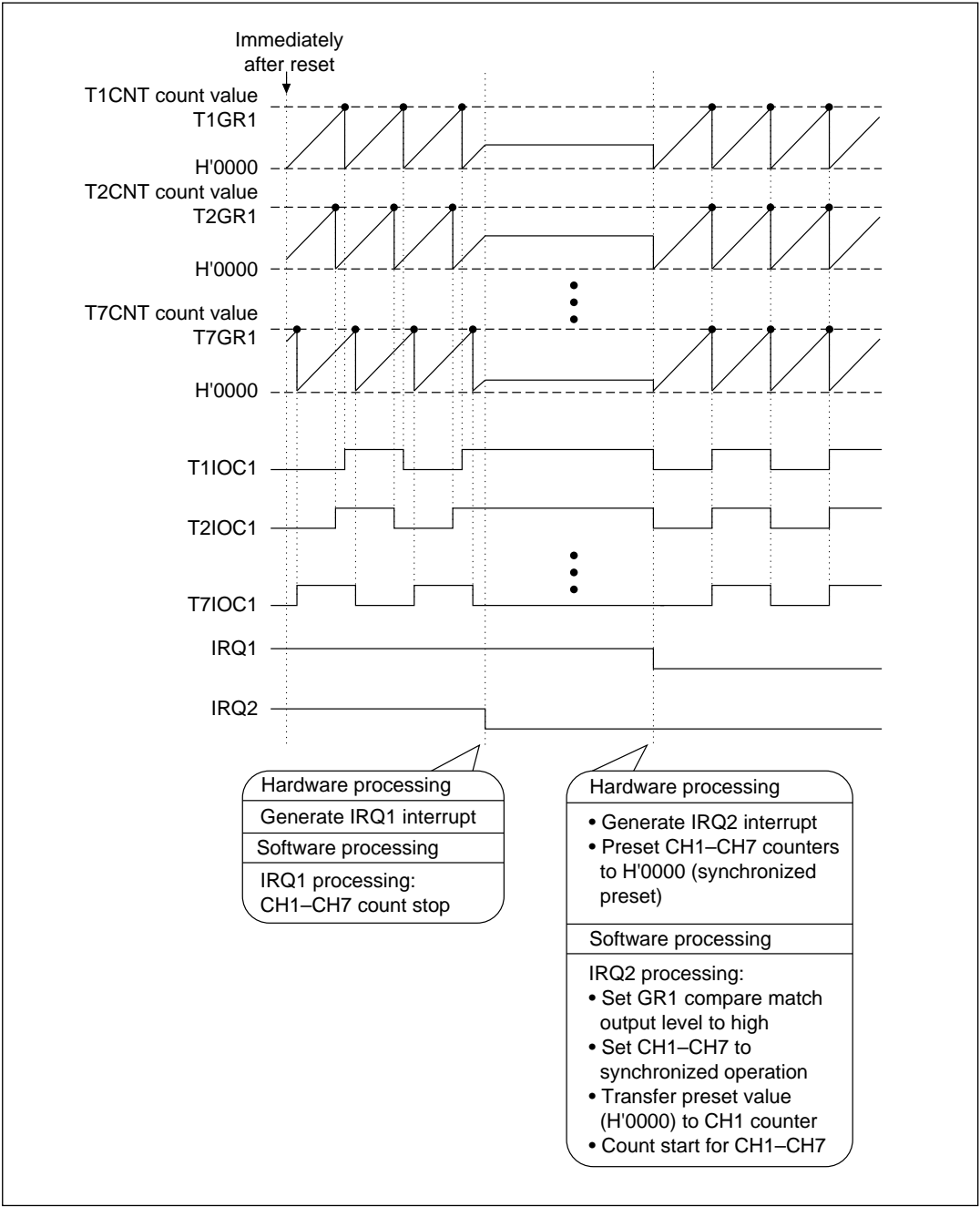


Figure 2.18 Operation of Counter Synchronization Preset and Simultaneous Start/Stop

2.4.4 Software

Tables 2.17 through 2.19 list information for this function.

Table 2.17 Modules

Module Name	Label Name	Function
Main routine	PRSTSPMN	Sets ch1—ch7 pulse output and interrupts.
IRQ1 interrupt	CNTSTP	Stops the ch1—ch7 timer counters.
IRQ2 interrupt	CNTPRST	Does synchronized presets and starts count of ch1—ch7 timer counters.
GR1 compare match interrupt	PLSINV1 — PLSINV7	Inverts ch1—ch7 pulse output.

Arguments:

This task example does not use any arguments.

Table 2.18 Internal Registers Used

Register Name	Function	Name of Module Used
T1CNT	ch1 timer counter	Main routine IRQ2 interrupt
T2CNT— T7CNT	ch2—ch7 timer counters.	Main routine
T1CRL— T7CRL	Selects timer counter clear sources.	Main routine
T1GR1— T7GR1	Sets 1/2 the timer counter value of the ch1—ch7 output pulse cycle.	Main routine
T1OER— T7OER	Selects compare match signal output level.	Main routine GR1 compare match interrupt
T1SRH— T7SRH	Selects enable/disable of compare match interrupt	Main routine
TSTR	Starts and stops the ch1—ch7 timer counters.	Main routine IRQ1 interrupt IRQ2 interrupt
TMDRA	Selects to make the timer counters operate in sync.	Main routine

Table 2.19 General Registers Used

Name of Module Used	Name of Register	Function
Main routine	R0	Used as work space when setting data.

RAM:

This task example does not use RAM except for arguments.

2.4.5 Flowcharts

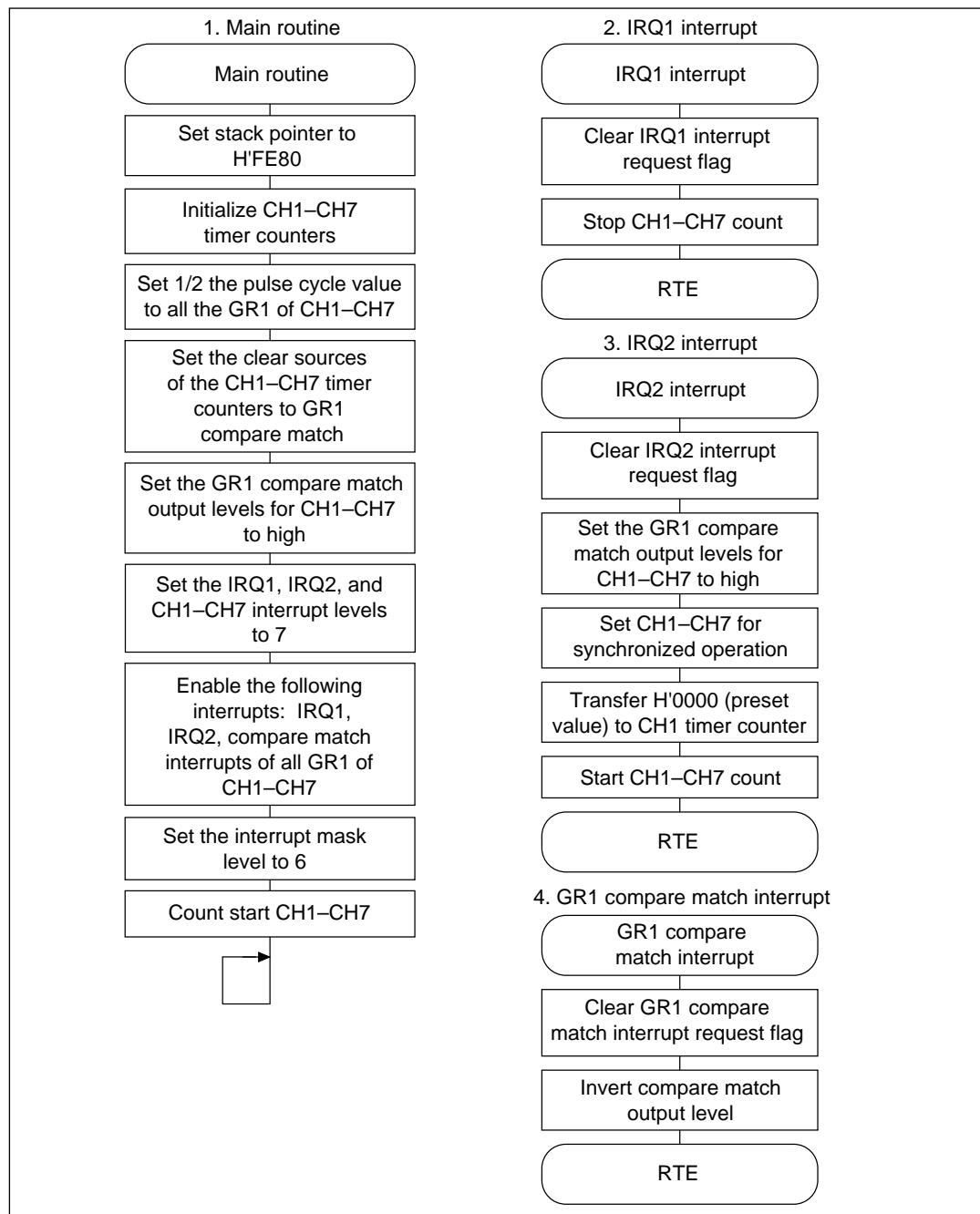


Figure 2.19 Timer Counter Synchronized Preset and Simultaneous Count Start/Stop Flowchart

2.5 5-Phase Pulse Output Upon External Trigger

Functions used: IPU

2.5.1 Specifications

As shown in figure 2.20, pulses of any pulse width are output from ch1—ch3, ch6 and ch7 at any delay times from the rise edge of an external pulse input to T1IOC1.

The delay time from the rise edge of the external pulse and the pulse width can be varied within the following ranges: delay time + 400 ns \leq delay time < external pulse cycle - pulse width; 100 ns \leq pulse width < external pulse cycle - delay time.

When operating at 10 MHz, the external pulse cycle can be set anywhere between 500 ns and 6.55 ms.

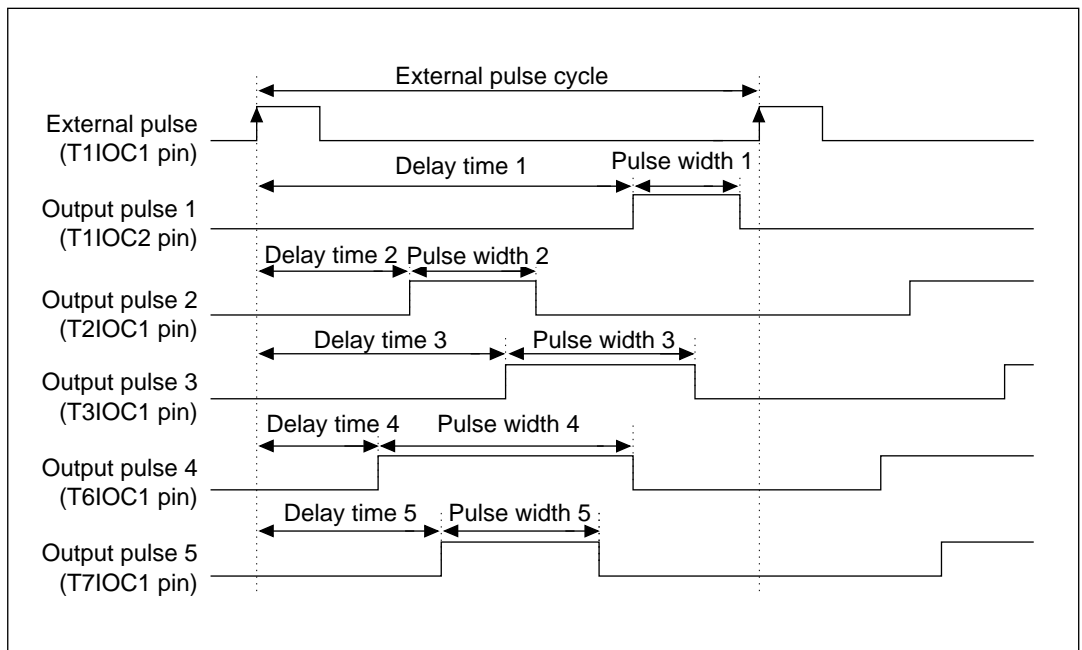


Figure 2.20 5-Phase Pulse Output from External Trigger

2.5.2 Functions Used

This task example uses the following functions of the IPU:

- Function that uses any of ch1—ch3, ch6 and ch7 as PWM timers (PWM mode)
- Function for clearing multiple timer counters simultaneously (synchronized clear)

Figure 2.21 is a block diagram of PWM output (ch1 only); figure 2.22 is a block diagram of a synchronized clear.

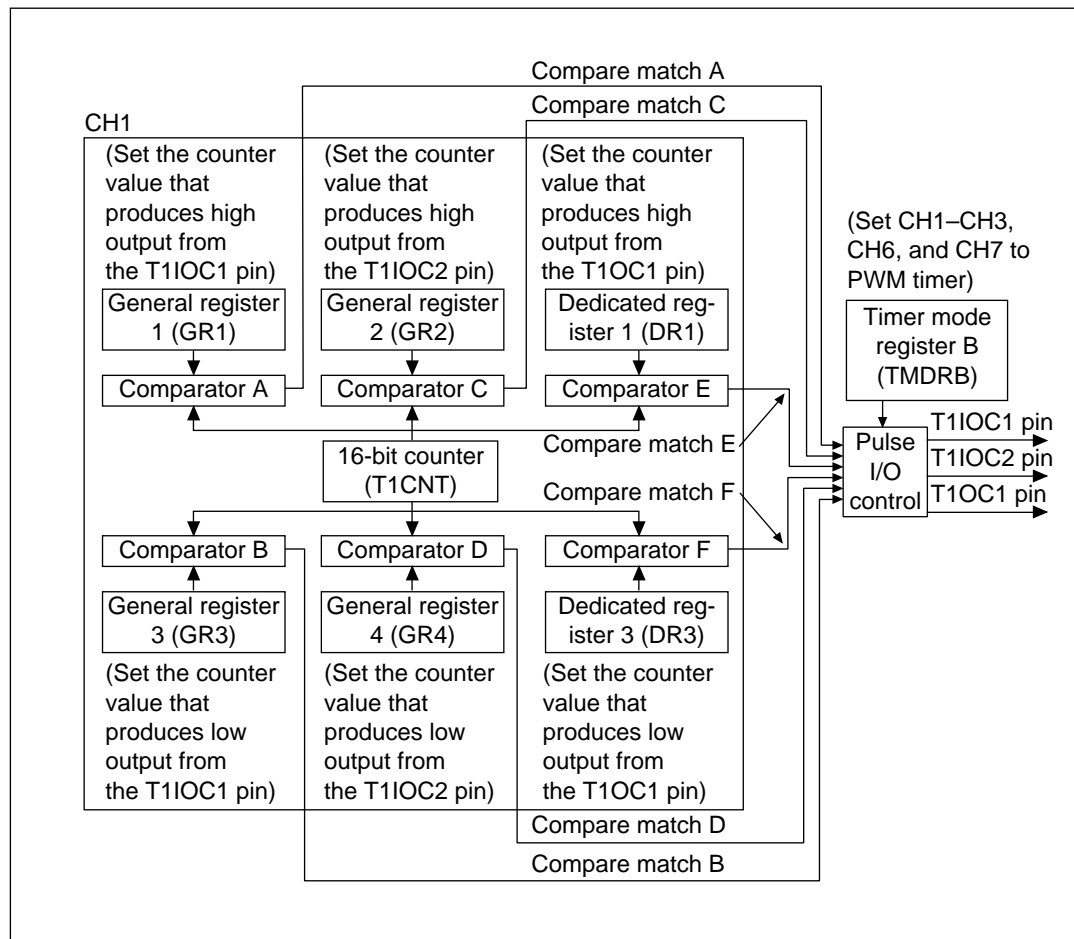


Figure 2.21 PWM Output Block Diagram

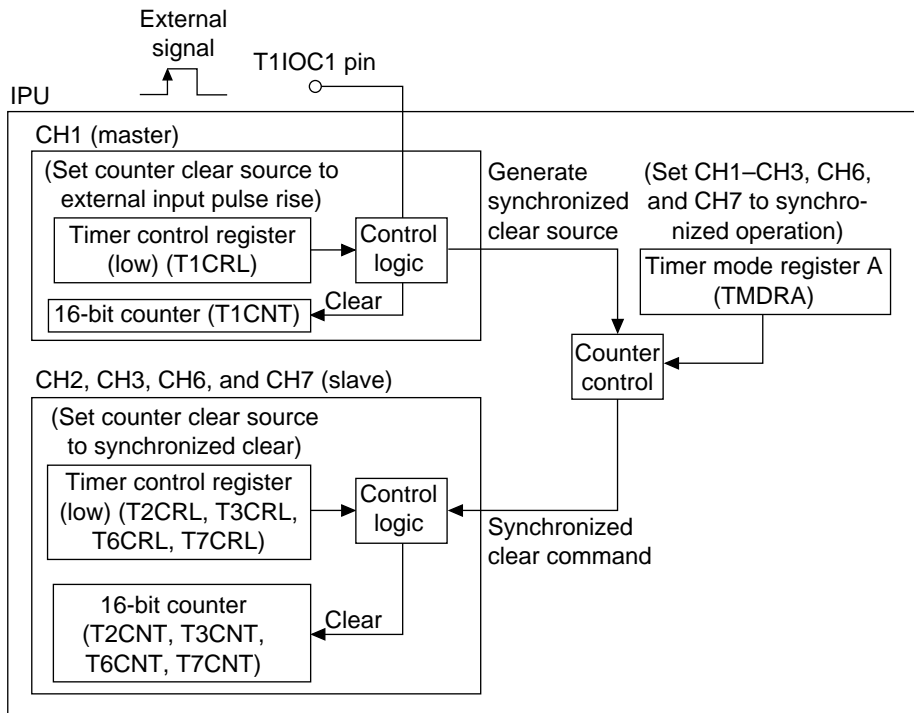


Figure 2.22 Block Diagram for Synchronized Clear Upon External Signal Rise

Table 2.20 lists the function allocations for this task example.

Table 2.20 IPU Function Allocation

IPU Function	Function
TMDRA	Selects channels to make operate in sync.
T1CRL—T3CRL, T6CRL, T7CRL	Selects timer counter clear sources.
TMDRB	Selects channels to operate as PWM timers.
T1GR2	Sets the T1IOC2 output level to high.
T1GR4	Sets the T1IOC2 output level to low.
T2GR1	Sets the T2IOC1 output level to high.
T2DR1	Sets the T2IOC1 output level to low.
T3GR1	Sets the T3IOC1 output level to high.
T3DR1	Sets the T3IOC1 output level to low.
T6GR1	Sets the T6IOC1 output level to high.
T6GR2	Sets the T6IOC1 output level to low.
T7GR1	Sets the T7IOC1 output level to high.
T7GR2	Sets the T7IOC1 output level to low.
TSTR	Starts and stops ch1—ch7 timer counters.

2.5.3 Operation

Figure 2.23 shows the principles of operation. Pulses of any pulse width are output from ch1—ch3, ch6 and ch7 at any delay times from the rise edge of an reference pulse input to T1IOC1 by H8/538 hardware processing and software processing.

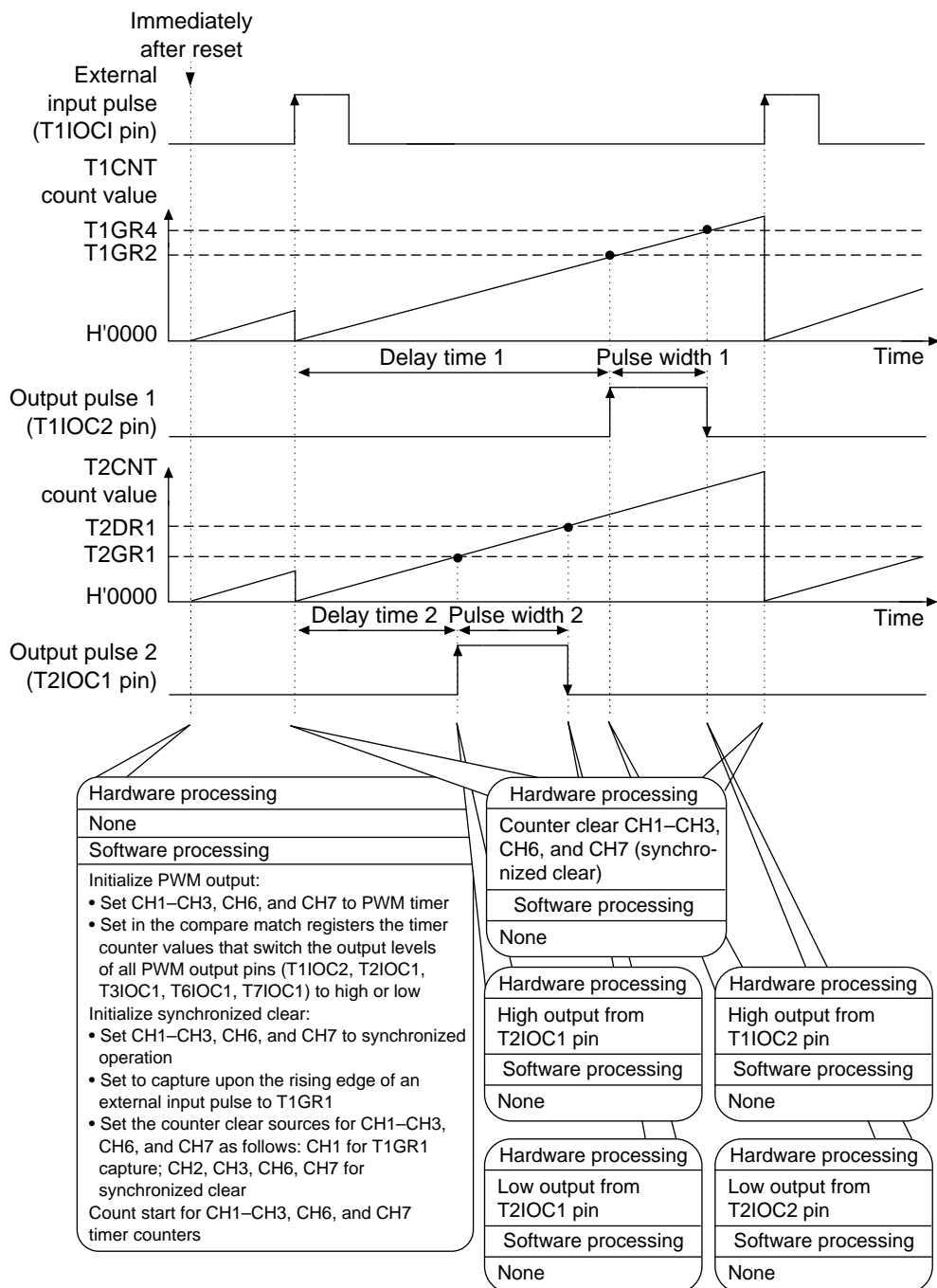


Figure 2.23 Operation for 5-Phase Pulse Output from External Trigger

2.5.4 Software

Tables 2.21 through 2.24 list software information for this function.

Table 2.21 Modules

Module Name	Label Name	Function
Main routine	EXT5POUTMN	Initializes for synchronized operation of ch1—ch3, ch6, ch7, and PWM output.

Table 2.22 Arguments

Label Name	Function	Data Length	Name of Module Used	I/O
PLSWIDTH1—5	Sets the timer counter value corresponding to High width of the pulse. The pulse High width is found by the following equation. Pulse High width (ns) = Timer counter value × System clock cycle (100 ns when operating at 10 MHz) × Input clock division ratio for each channel.	2 bytes	Main routine	I
PLSDLY1—5	Sets the timer value corresponding to the delay time from the rise of the external input pulse. The delay time is found by the following equation. Delay time (ns) = Timer counter value × System clock cycle (100 ns when operating at 10 MHz) × Input clock division ratio for each channel.	2 bytes	Main routine	I

Table 2.23 Internal Registers Used

Register Name	Function	Module Used
TMDRA	Selects channels to operate in sync.	Main routine
T1CRL—T3CRL, T6CRL, T7CRL	Selects timer counter clear sources.	
TMDRB	Selects channels to operate as PWM timers.	
T1GR2	Sets the timer counter value that makes T1IOC2 output high.	
T1GR4	Sets the timer counter value that makes T1IOC2 output low.	
T2GR1	Sets the timer counter value that makes T2IOC1 output high.	
T2DR1	Sets the timer counter value that makes T2IOC1 output low.	
T3GR1	Sets the timer counter value that makes T3IOC1 output high.	
T3DR1	Sets the timer counter value that makes T3IOC1 output low.	
T6GR1	Sets the timer counter value that makes T6IOC1 output high.	
T6GR2	Sets the timer counter value that makes T6IOC1 output low.	
T7GR1	Sets the timer counter value that makes T7IOC1 output high.	
T7GR2	Sets the timer counter value that makes T7IOC1 output low.	
TSTR	Starts and stops ch1—ch7 as timer counters.	

Table 2.24 General Registers Used

Name of Module Used	Name of Register	Function
Main routine	R0—R1	Used as work space when setting data.

RAM:

This task example does not use RAM except for arguments.

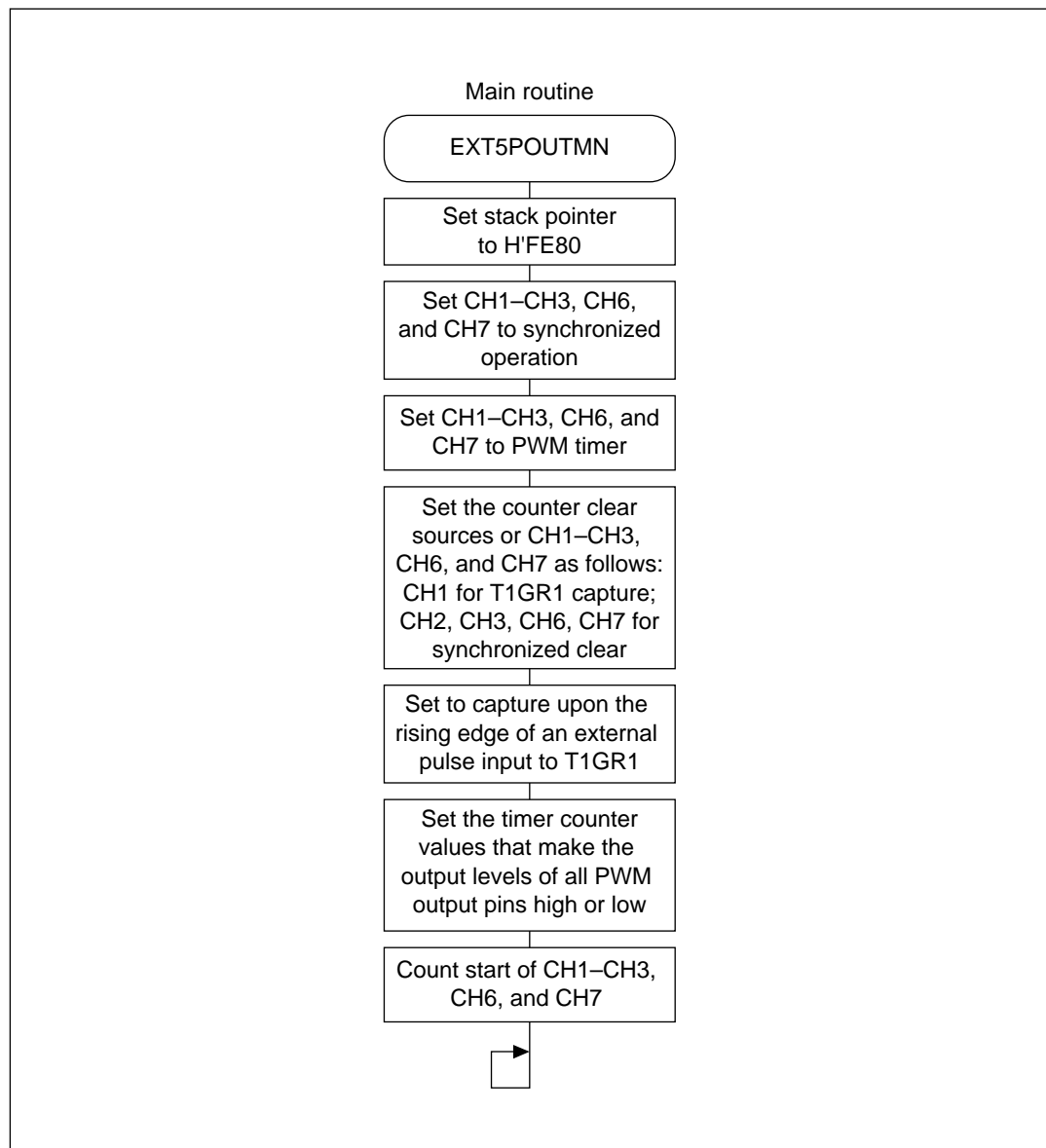


Figure 2.24 5-Phase Pulse Output Flowchart

2.6 PWM Output 2

Functions used: PWM output mode, synchronized clear

2.6.1 Specifications

As shown in figure 2.25, a PWM waveform (9 phase) is output for use in generating a equilateral triangular waveform. When operating at 10 MHz, the PWM cycle can be set anywhere between 2 μ s and 6.55 ms.

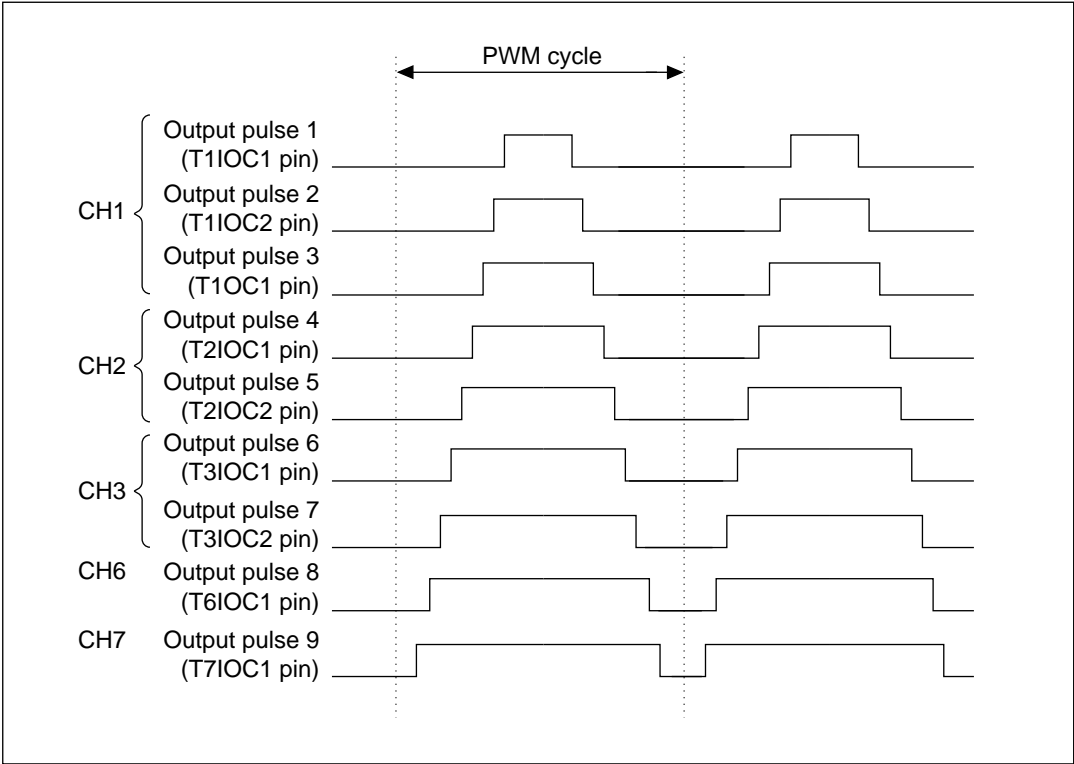


Figure 2.25 9-Phase PWM Output

2.6.2 Functions Used

This task example uses the following functions of the IPU:

- Function that uses ch1—ch3, ch6 and ch7 as PWM timers (PWM mode)
- Function for clearing multiple timer counters simultaneously (synchronized clear)

Figure 2.26 is a block diagram of PWM output (ch1 only); figure 2.27 is a block diagram of a synchronized clear.

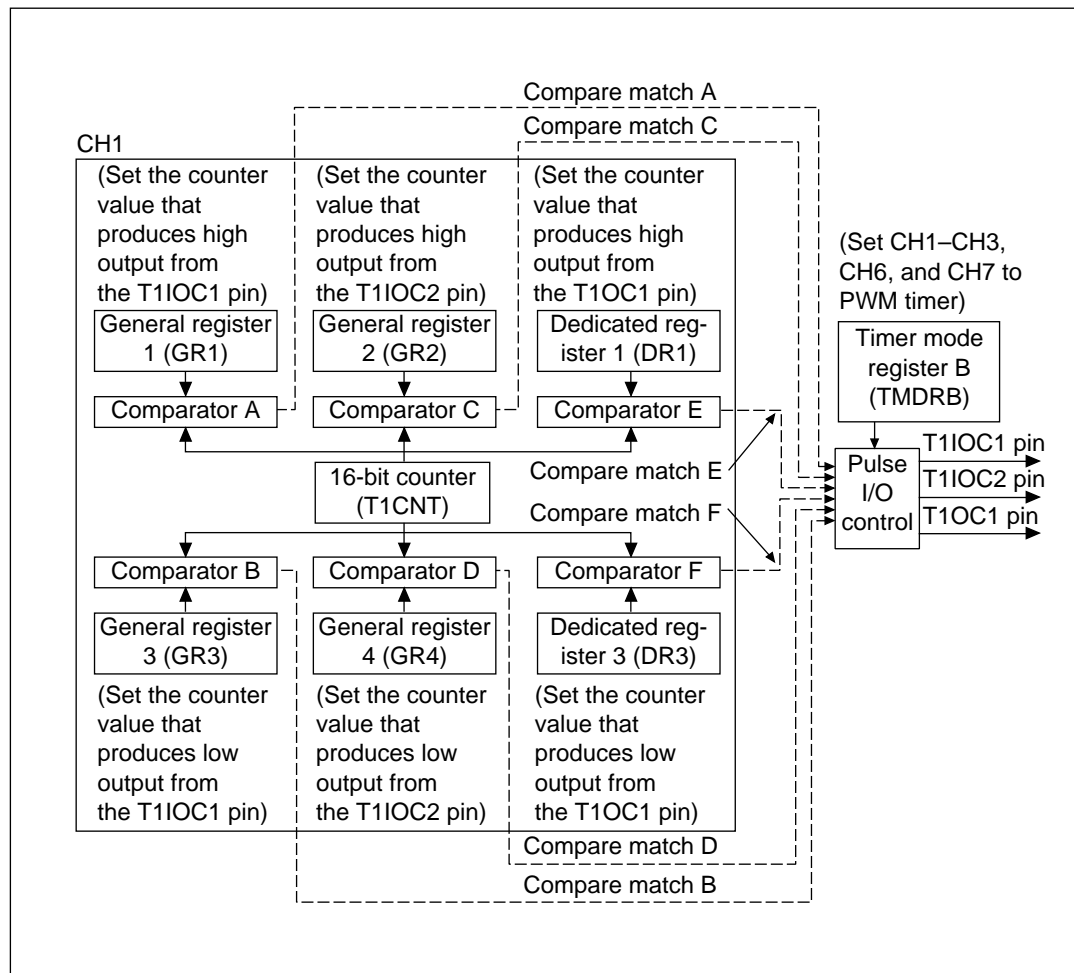


Figure 2.26 PWM Output Block Diagram

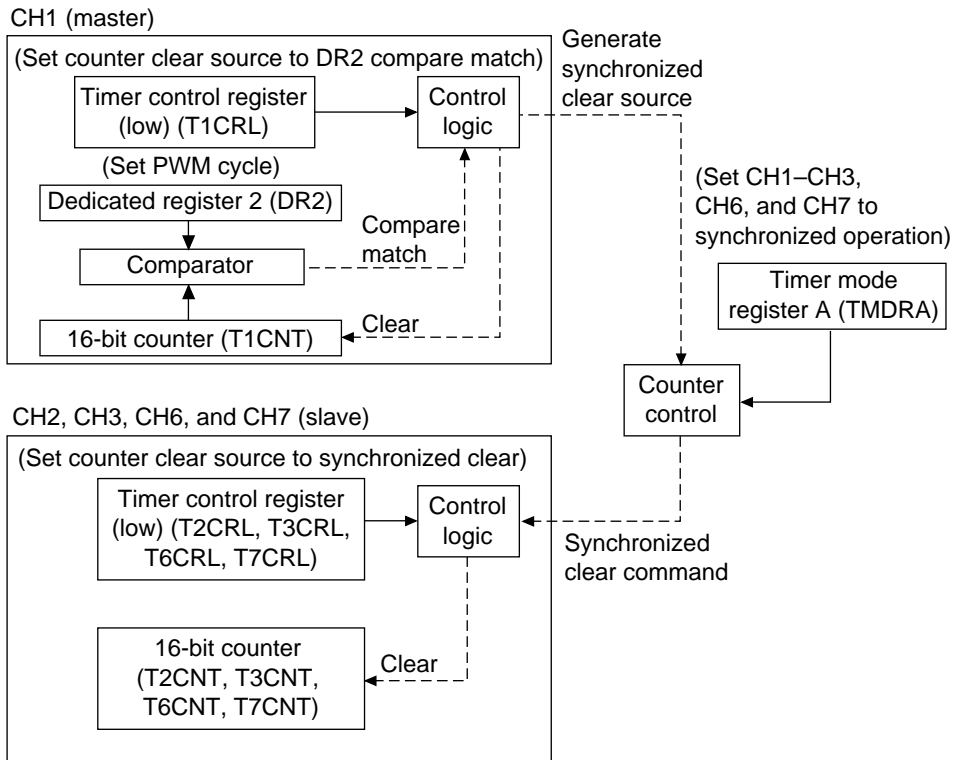


Figure 2.27 Synchronized Clear Block Diagram

Table 2.25 lists the function allocations for this task example. IPU functions are allocated as listed in table 2.25, and PWM pulses are output.

Table 2.25 IPU Function Allocation

IPU Function	Function
T1IOC1—3, T2IOC1—2, T3IOC1—2, T6IOC1, T7IOC1	Pins for PWM pulse output.
TMDRA	Selects channels to operate in sync.
T1CRL—T3CRL, T6CRL, T7CRL	Selects timer counter clear sources.
TMDRB	Selects channels to operate as PWM timers.
T1DR2	Sets the PWM cycle.
T1GR1	Sets the timer counter value that makes T1IOC1 output high.
T1GR3	Sets the timer counter value that makes T1IOC1 output low.
T1GR2	Sets the timer counter value that makes T1IOC2 output high.
T1GR4	Sets the timer counter value that makes T1IOC2 output low.
T1DR1	Sets the timer counter value that makes T1IOC1 output high.
T1DR3	Sets the timer counter value that makes T1IOC1 output low.
T2GR1	Sets the timer counter value that makes T2IOC1 output high.
T2DR1	Sets the timer counter value that makes T2IOC1 output low.
T2GR2	Sets the timer counter value that makes T2IOC2 output high.
T2DR2	Sets the timer counter value that makes T2IOC2 output low.
T3GR1	Sets the timer counter value that makes T3IOC1 output high.
T3DR1	Sets the timer counter value that makes T3IOC1 output low.
T3GR2	Sets the timer counter value that makes T3IOC2 output high.
T3DR2	Sets the timer counter value that makes T3IOC2 output low.
T6GR1	Sets the timer counter value that makes T6IOC1 output high.
T6GR2	Sets the timer counter value that makes T6IOC1 output low.
T7GR1	Sets the timer counter value that makes T7IOC1 output high.
T7GR2	Sets the timer counter value that makes T7IOC1 output low.
TSTR	Starts and stops ch1—ch3, ch6 and ch7 timer counters.

2.6.3 Operation

Figure 2.28 shows the principles of operation. A 9-phase PWM waveform is output for use in generating an equilateral triangular waveform from the PWM output pins of ch1—ch3, ch6 and ch7 (everything except T1IOC1, T2IOC1 and T3IOC1 is omitted) by H8/538 hardware processing and software processing as shown in figure 2.28.

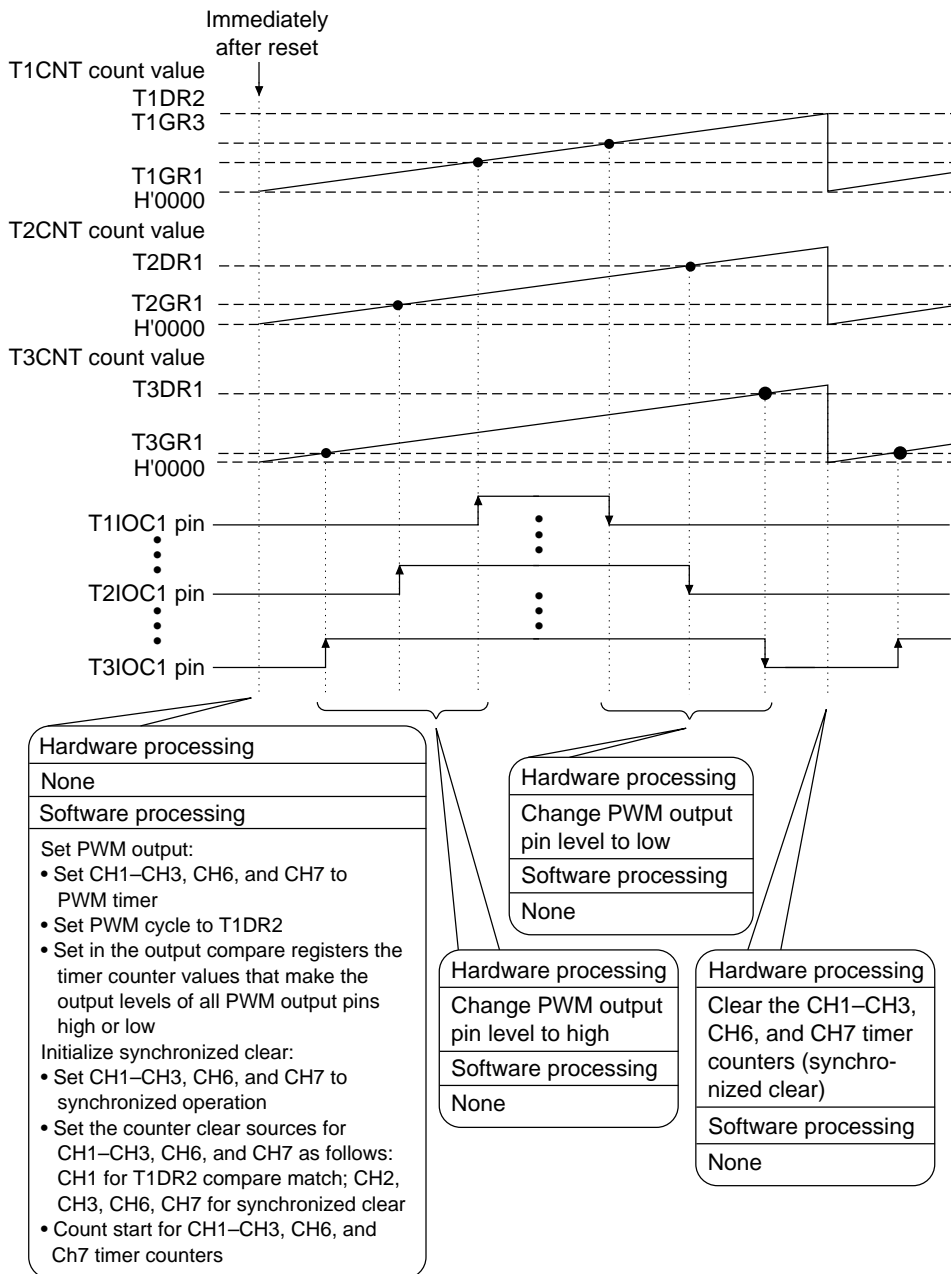


Figure 2.28 PWM Waveform (9-Phase) Used to Generate an Equilateral Triangle Waveform

2.6.4 Software

Tables 2.26 through 2.29 list software information for this function.

Table 2.26 Modules

Module Name	Label Name	Function
Main routine	PWMOUT2	Sets for synchronized clear and PWM output for ch1—ch3, ch6 and ch7.

Table 2.27 Arguments

Label Name	Function	Data Length	Name of Module Used	I/O
PWM_HI1 — PWM_H19	Sets the timer counter value corresponding to High width of the pulse. The pulse High width is found by the following equation. Pulse High width (ns) = Timer counter value × System clock cycle (100 ns when operating at 10 MHz) × Input clock division ratio for each channel.	2 bytes	Main routine	I
PWM_CYC	Sets the timer value corresponding to PWM cycle. The PWM cycle is found by the following equation. PWM cycle (ns) = Timer counter value × System clock cycle (100 ns when operating at 10 MHz) × Input clock division ratio for each channel.	2 bytes	Main routine	I

Table 2.28 Internal Registers Used

Register Name	Function	Module Used
TMDRA	Selects channels to operate in sync.	Main routine
T1CRL—T3CRL, T6CRL, T7CRL	Selects timer counter clear sources.	
TMDRB	Selects channels to operate as PWM timers.	
T1DR2	Sets the PWM cycle.	
T1GR1	Sets the timer counter value that makes T1IOC1 output high.	
T1GR3	Sets the timer counter value that makes T1IOC1 output low.	
T1GR2	Sets the timer counter value that makes T1IOC2 output high.	
T1GR4	Sets the timer counter value that makes T1IOC2 output low.	
T1DR1	Sets the timer counter value that makes T1IOC1 output high.	
T1DR3	Sets the timer counter value that makes T1IOC1 output low.	
T2GR1	Sets the timer counter value that makes T2IOC1 output high.	
T2DR1	Sets the timer counter value that makes T2IOC1 output low.	
T2GR2	Sets the timer counter value that makes T2IOC2 output high.	
T2DR2	Sets the timer counter value that makes T2IOC2 output low.	
T3GR1	Sets the timer counter value that makes T3IOC1 output high.	
T3DR1	Sets the timer counter value that makes T3IOC1 output low.	
T3GR2	Sets the timer counter value that makes T3IOC2 output high.	
T3DR2	Sets the timer counter value that makes T3IOC2 output low.	
T6GR1	Sets the timer counter value that makes T6IOC1 output high.	
T6GR2	Sets the timer counter value that makes T6IOC1 output low.	
T7GR1	Sets the timer counter value that makes T7IOC1 output high.	
T7GR2	Sets the timer counter value that makes T7IOC1 output low.	
TSTR	Starts and stops ch1—ch7 as timer counters.	

Table 2.29 General Registers Used

Name of Module Used	Name of Register	Function
Main routine	R0—R2	Used as work space when setting data.

RAM:

This task example does not use RAM except for arguments.

2.6.5 Flowcharts

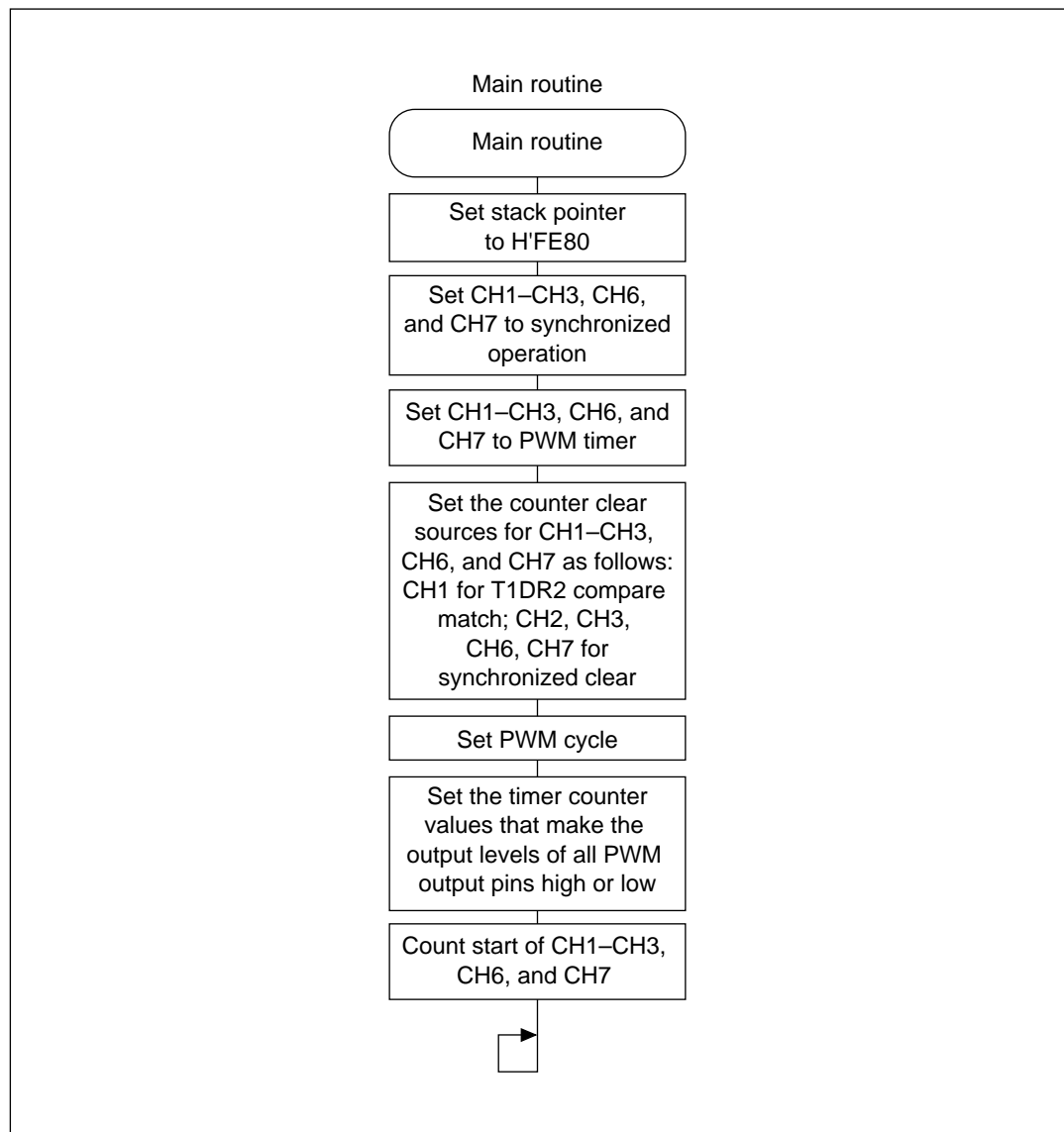


Figure 2.29 PWM Output 2 Flowchart

2.7 2-Phase Encoder Count

Functions used: IPU

2.7.1 Specifications

As shown in figure 2.30, the phase differential of a 2-phase encoder output pulse input to an external clock pin (TCLK1, TCLK2) is detected and the number of counts counted up or down during the measurement period is set in RAM.

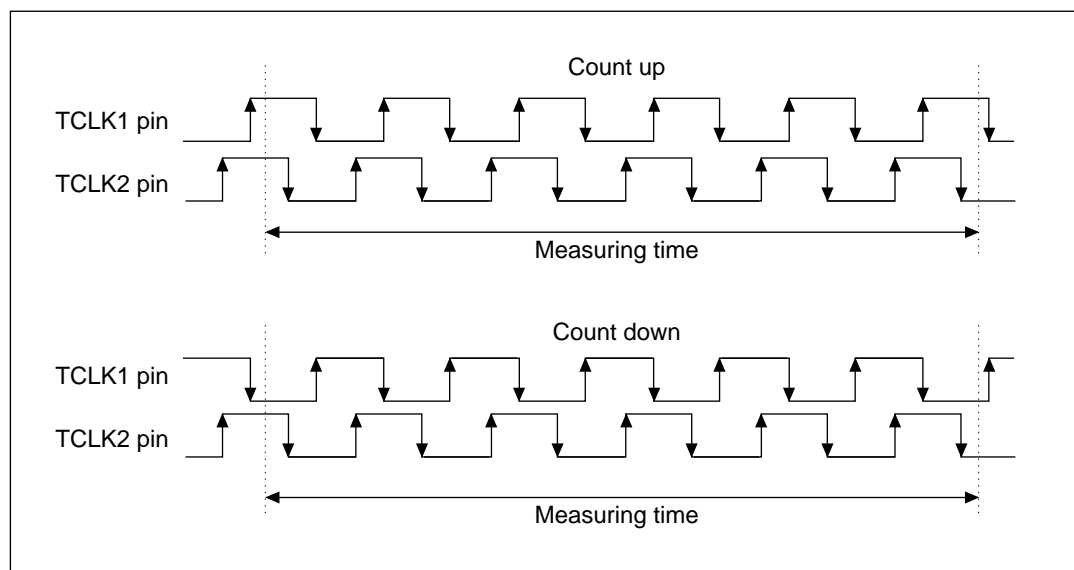


Figure 2.30 2-Phase Encoder Count

2.7.2 Functions Used

This task example uses the following functions of the IPU to do a 2-phase encoder count. Figure 2.31 is a block diagram of a 2-phase encoder count.

- Function for detecting the phase differential of a 2-phase encoder pulse input to an external clock pin (TCLK1, TCLK2) and counting the ch7 counter up or down (phase counting mode).
- Function for transferring the value of the counter operating on an external clock to the general register upon a compare match with another channel (program cycle counting mode).

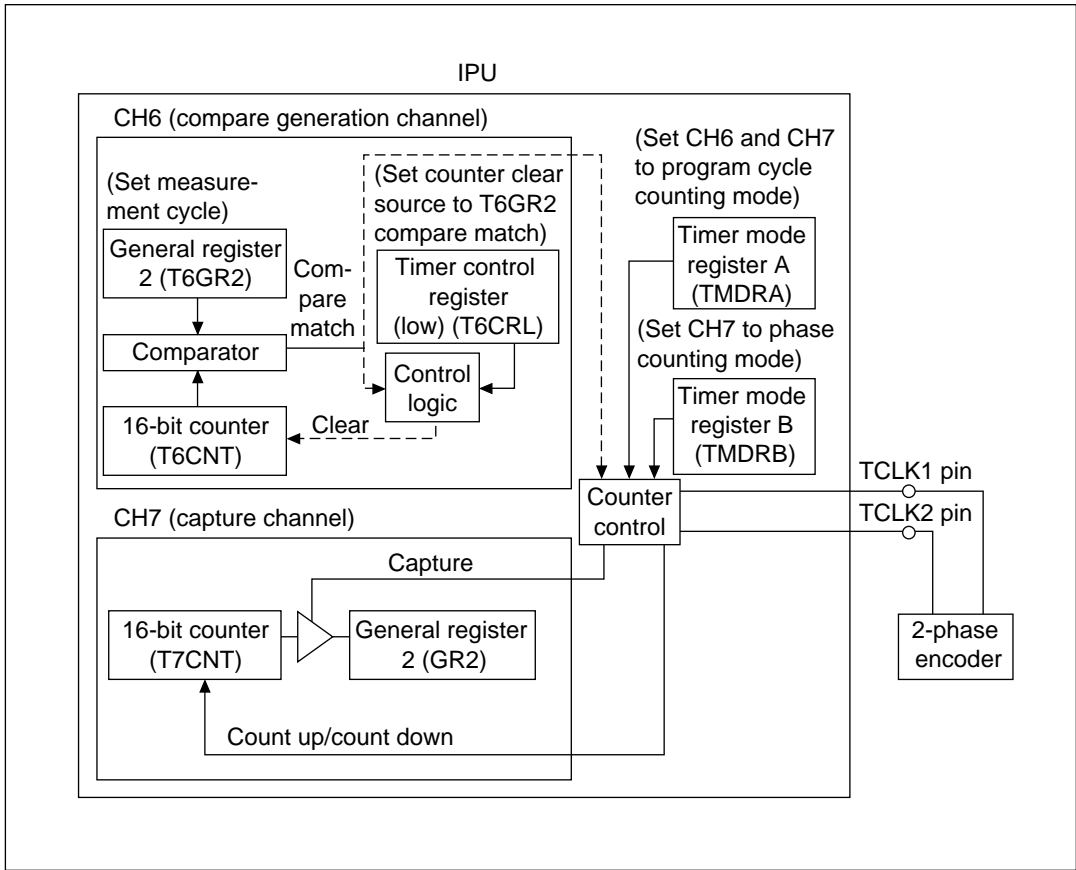


Figure 2.31 Block Diagram of 2-Phase Encoder Count

Table 2.30 lists the function allocations for this task example. IPU functions are allocated as shown in table 2.30, and a 2-phase encoder count performed.

Table 2.30 IPU Function Allocation

IPU Function	Function
TCLK1, 2	Input pin for 2-phase encoder pulse.
T6GR2	Sets the measurement time.

2.7.3 Operation

Figure 2.32 shows the principles of operation. A 2-phase encoder count (count up) is done by H8/538 hardware processing and software processing. Figure 2.33 shows the principles of operation. A 2-phase encoder count (count down) is done by H8/538 hardware processing and software processing.

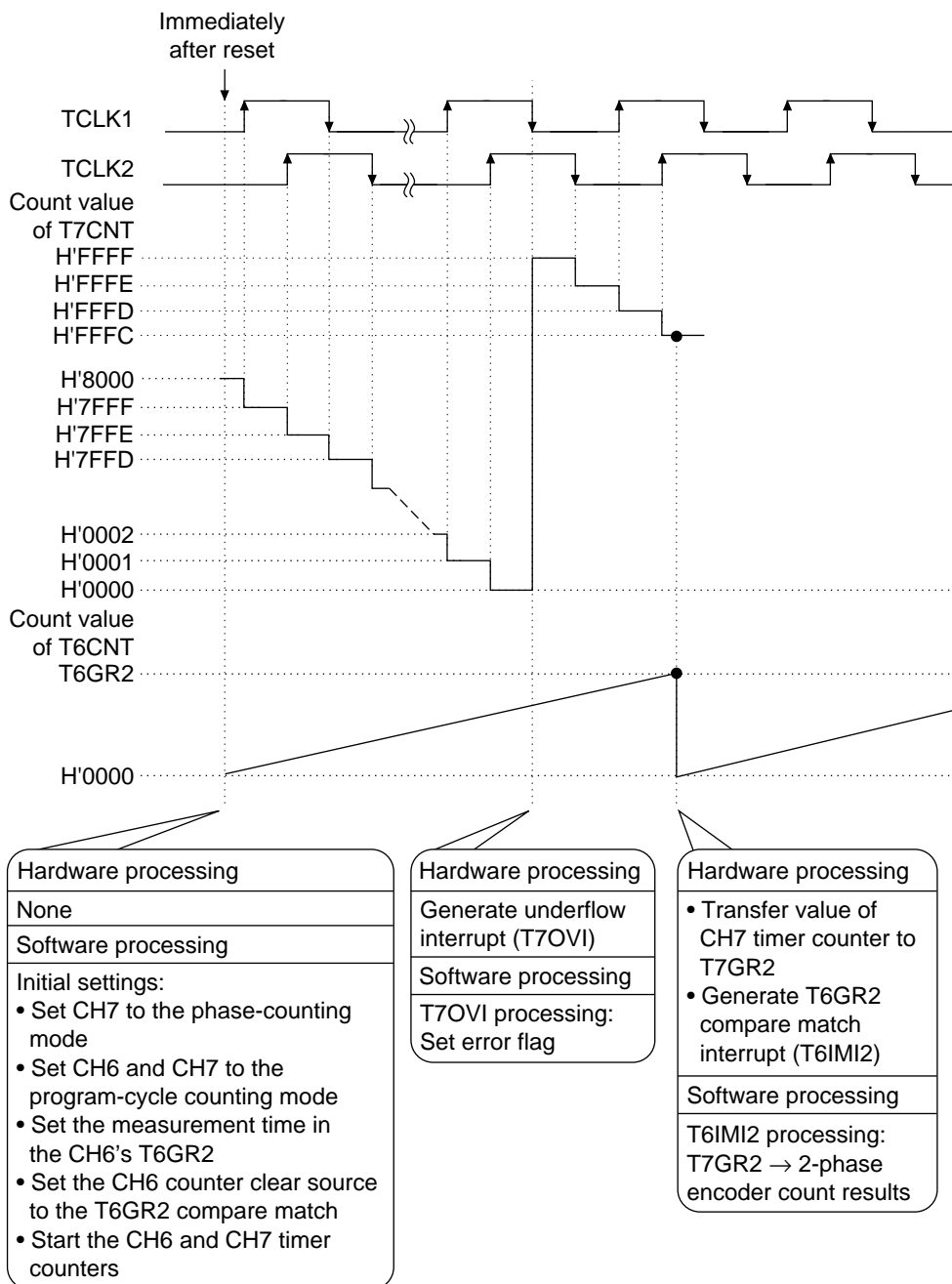
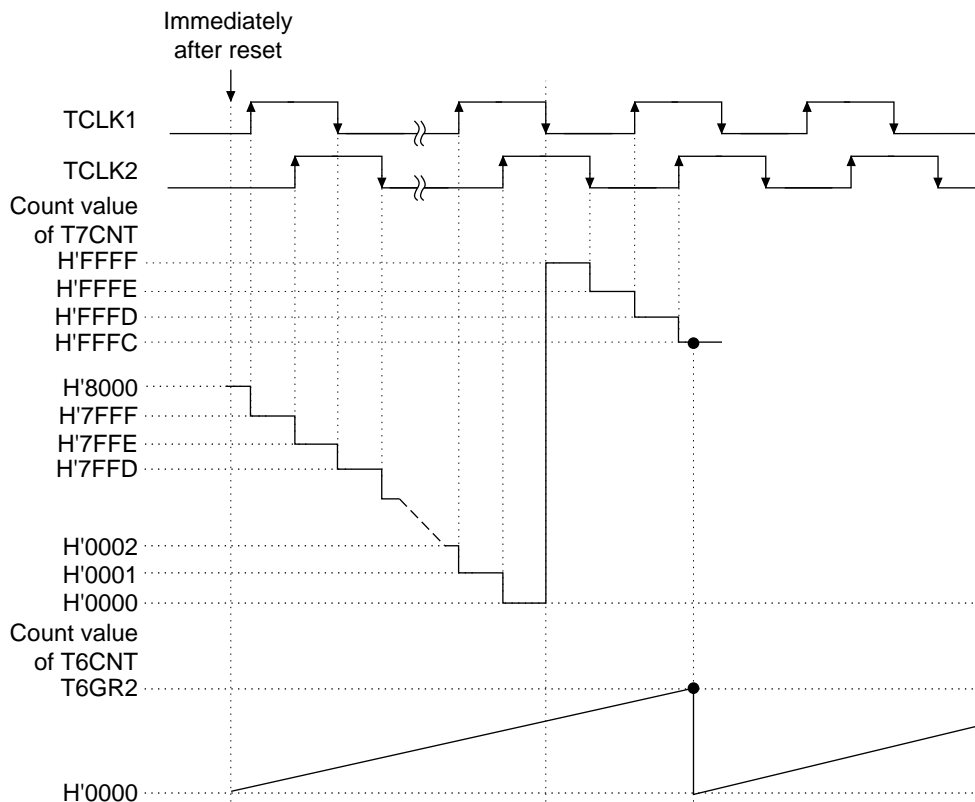


Figure 2.32 Operation of 2-Phase Encoder Count (Upcount)



Hardware processing	Hardware processing	Hardware processing
None	Generate underflow interrupt (T7OVI)	<ul style="list-style-type: none"> Transfer value of CH7 timer counter to T7GR2 Generate T6GR2 compare match interrupt (T6IMI2)
Software processing	Software processing	Software processing
Initial settings: <ul style="list-style-type: none"> Set CH7 to the phase-counting mode Set CH6 and CH7 to the program-cycle counting mode Set the measurement time in the CH6's T6GR2 Set the CH6 counter clear source to the T6GR2 compare match Start the CH6 and CH7 timer counters 	T7OVI processing: <ul style="list-style-type: none"> Set error flag 	T6IMI2 processing: <ul style="list-style-type: none"> T7GR2 → 2-phase encoder count results

Figure 2.33 Operation of 2-Phase Encoder Count (Downcount)

2.7.4 Software

Tables 2.31 through 2.34 list software information for this function.

Table 2.31 Modules

Module Name	Label Name	Function
Main routine	CNTMN	Initializes 2-phase encoder count.
Compare match interrupt	RAMSET	Stores the resulting count in RAM.
Overflow/underflow interrupt	FLGSET	Sets the error flag.

Table 2.32 Arguments

Label Name	Function	Data Length	Name of Module Used	I/O
CNT_RSLT	Sets the results of counting during the measurement time.	2 bytes	Compare match interrupt	O
ERRFLG	Indicates whether an overflow or underflow has occurred. 1: Occurred; 0: None.	1 bit	Overflow/underflow interrupt	O
CNT_TIM	Sets the measurement time.	2 bytes	Main routine	I

Table 2.33 Internal Registers Used

Register Name	Function	Name of Module Used
TMDRA	Selects the channel pair to be operated in cycle-counting mode.	Main routine
TMDRB	Selects ordinary operation/phase-counting mode operation for ch7.	Main routine
T6CRH	Selects the ch6 operating clock.	Main routine
T6GR2	Sets the measurement time.	Main routine
T7GR2	Sets the ch7 timer counter value at T6GR2 compare match.	Compare match interrupt
TSTR	Starts or stops the count of ch1—ch7 timer counters.	Main routine

Table 2.34 General Registers Used

Name of Module Used	Name of Register	Function
Main routine	R0	Used as work space when setting data.

RAM:

This task example does not use RAM except for arguments.

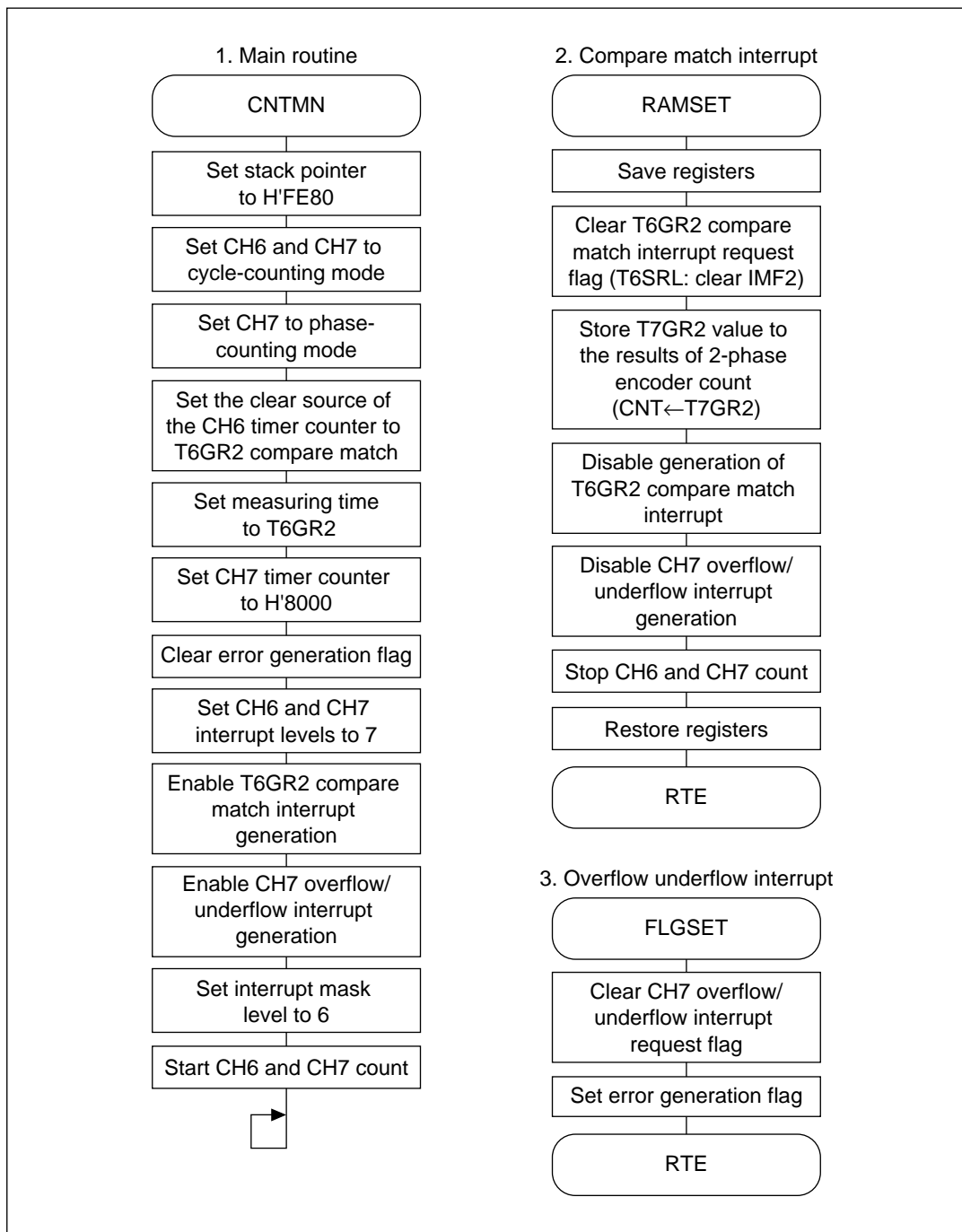


Figure 2.34 2-Phase Encoder Count Flowchart

2.8 2-Phase Encoder Count and 7-Phase PWM Output

Functions used: IPU

2.8.1 Specifications

As shown in figure 2.35, PWM output (7-phase) is produced on ch1—ch3 for use in generating an equilateral triangular waveform and the counter value is found on ch4 and ch7 for the up-count or down-count of the 2-phase encoder during the measurement period. The measurement is started by the fall of IRQ1 and the result is set in RAM.

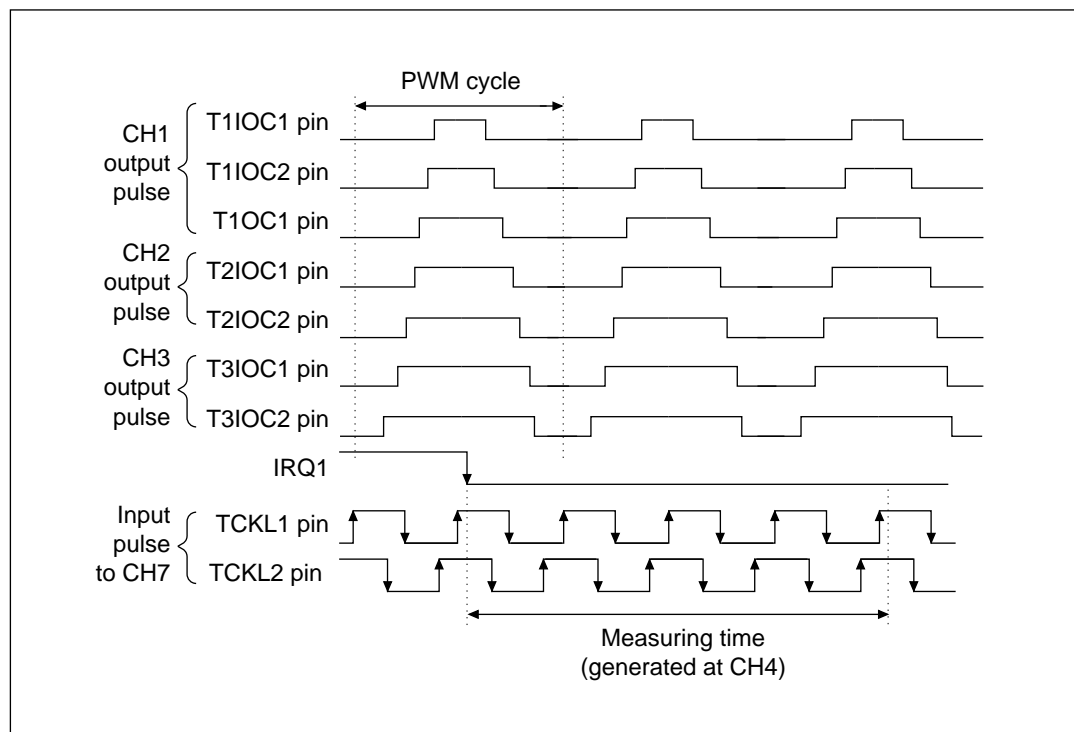


Figure 2.35 PWM Output (7-Phase) and 2-Phase Encoder Count

2.8.2 Functions Used

This task example uses the following functions of the IPU to produce PWM output (7-phase) for use in generating an equilateral triangular waveform and does a 2-phase encoder count:

- Function that uses any of ch1—ch3, ch6 and ch7 as PWM timers (PWM mode)
- Function for clearing multiple timer counters simultaneously (synchronized clear)
- Function for detecting the phase differential of a 2-phase encoder output pulse input to an external clock pin (TCLK1, TCLK2) and counting the ch7 timer up or down (phase counting mode)
- Function for transferring the value of the counter operating on an external clock to the general register upon a compare match with another channel (program cycle counting mode)

Figure 2.36 is a block diagram of the 2-phase encoder count, and figures 2.37 and 2.38 are block diagrams of PWM mode and synchronized clear.

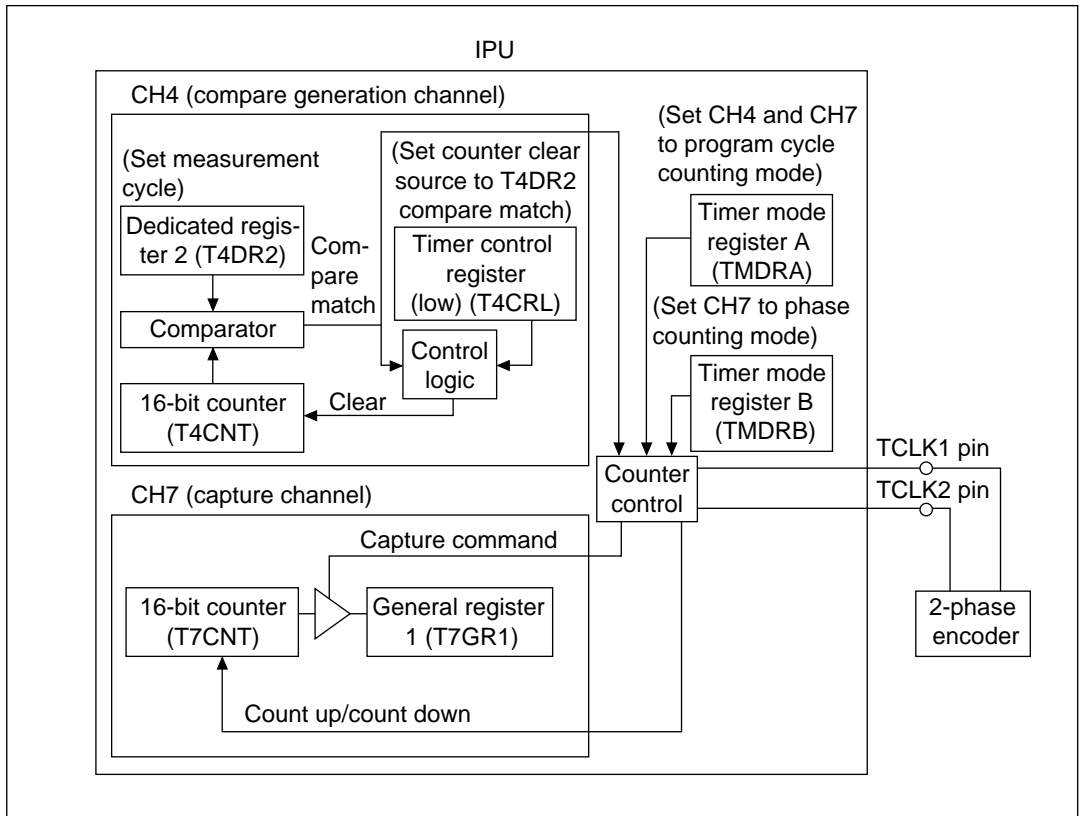


Figure 2.36 Block Diagram of 2-Phase Encoder Count

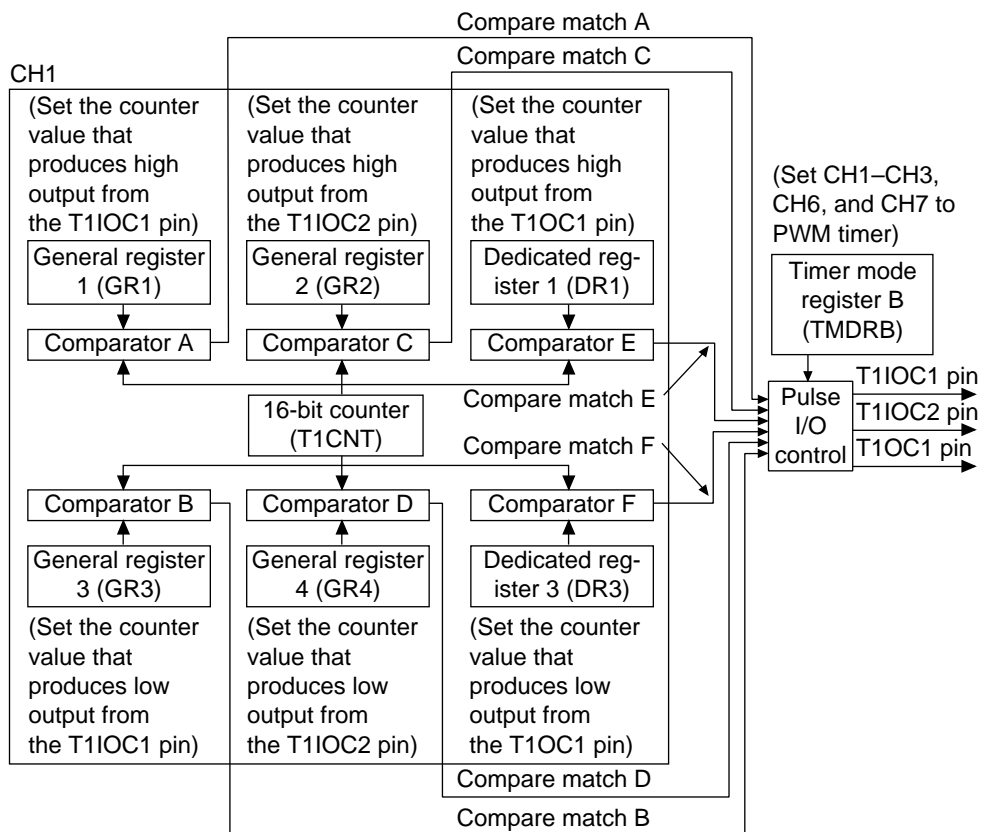


Figure 2.37 PWM Output Block Diagram

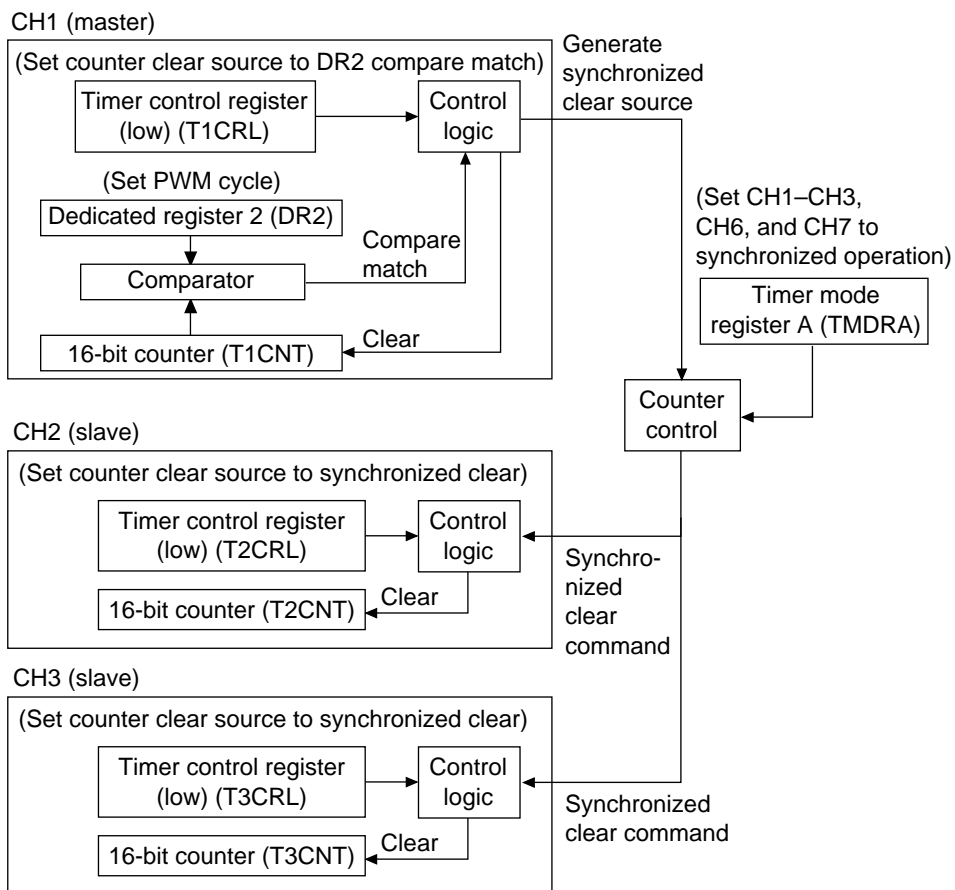


Figure 2.38 Synchronized Clear Block Diagram

Table 2.35 lists the function allocations for this task example. IPU functions are allocated as listed in table 2.35, and the cycle is measured.

Table 2.35 IPU Function Allocation

IPU Function	Function
TCLK1, 2	Input pin for 2-phase encoder pulse.
T1IOC1—3, T2IOC1—2, T3IOC1—2	PWM pulse output pin.
T7CNT	ch7 counter.
T4CRL	Selects ch4 counter clear source.
T4DR2	Sets the measurement time.
T7GR1	Sets the ch7 timer counter value at T4DR2 compare match.
T1CRL—T3CRL	Selects timer counter clear sources. ch1 (master): DR2 compare match (selects synchronized clear source); ch2—ch3 (slave): Synchronized clear
TMDRA	Synchronizes operation of ch1—ch3, and sets ch4 and ch7 to cycle-counting mode.
TMDRB	Makes ch1—ch3 operate as PWM timers, and sets ch7 to phase-counting mode.
T1DR2	Sets the PWM cycle.
T1GR1	Sets the timer counter value that makes T1IOC1 output high.
T1GR3	Sets the timer counter value that makes T1IOC1 output low.
T1GR2	Sets the timer counter value that makes T1IOC2 output high.
T1GR4	Sets the timer counter value that makes T1IOC2 output low.
T1DR1	Sets the timer counter value that makes T1IOC1 output high.
T1DR3	Sets the timer counter value that makes T1IOC1 output low.
T2GR1	Sets the timer counter value that makes T2IOC1 output high.
T2DR1	Sets the timer counter value that makes T2IOC1 output low.
T2GR2	Sets the timer counter value that makes T2IOC2 output high.
T2DR2	Sets the timer counter value that makes T2IOC2 output low.
T3GR1	Sets the timer counter value that makes T3IOC1 output high.
T3DR1	Sets the timer counter value that makes T3IOC1 output low.
T3GR2	Sets the timer counter value that makes T3IOC2 output high.
T3DR2	Sets the timer counter value that makes T3IOC2 output low.
TSTR	Starts and stops ch1—ch7 as timer counters.

2.8.3 Operation

Figure 2.39 shows the principles of operation. A 7-phase PWM waveform is output for use in generating an equilateral triangular waveform from the PWM output pins of ch1—ch3 (everything except T1IOC1, T2IOC1 and T3IOC1 is omitted) by H8/538 hardware processing and software processing. Figure 2.40 shows the principles of operation. A 2-phase encoder count (upcount) is done by H8/538 hardware processing and software processing. Figure 2.41 shows the principles of operation. A 2-phase encoder count (count down) is done by H8/538 hardware processing and software processing.

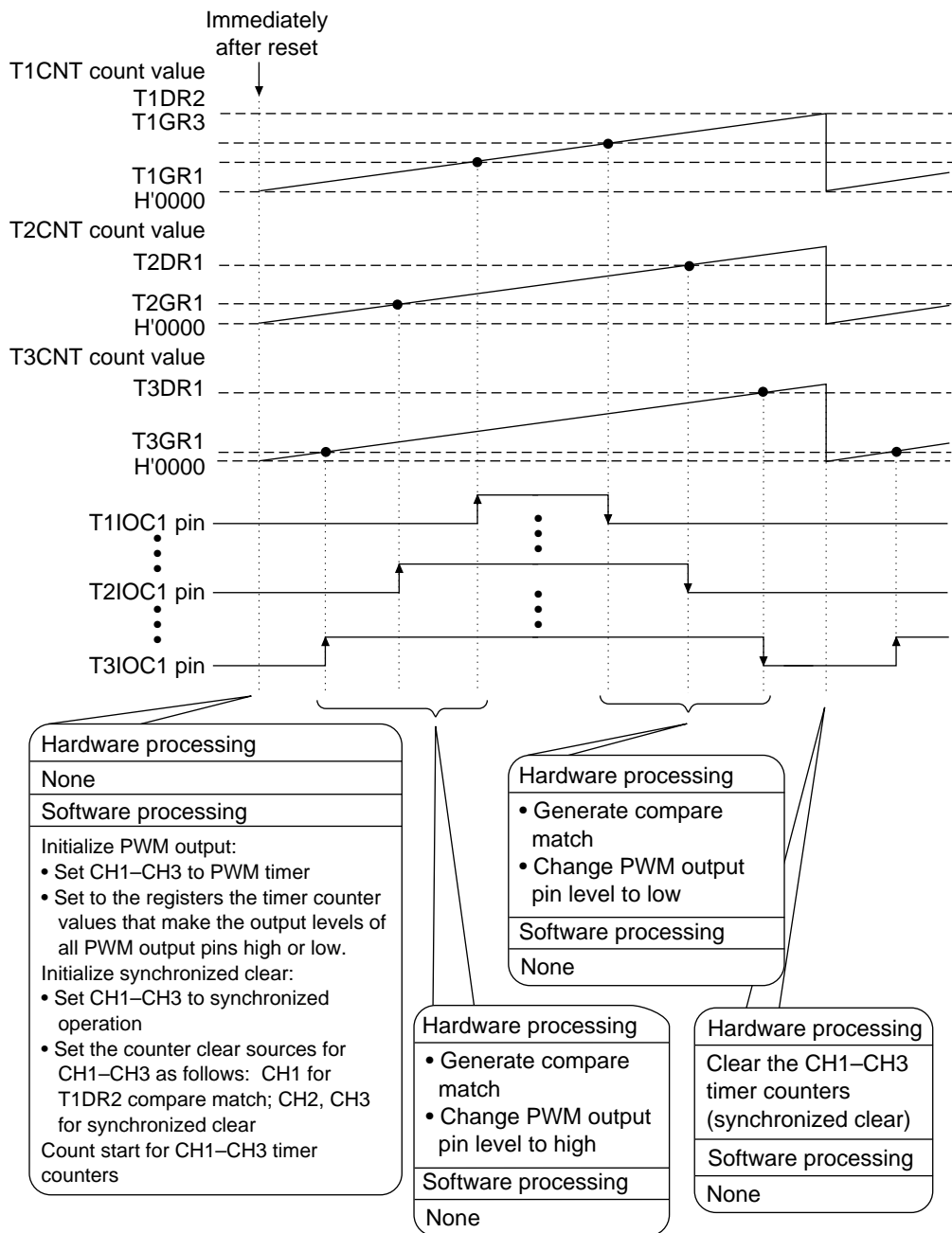


Figure 2.39 PWM Waveform (7-Phase) Used to Generate an Equilateral Triangle Waveform

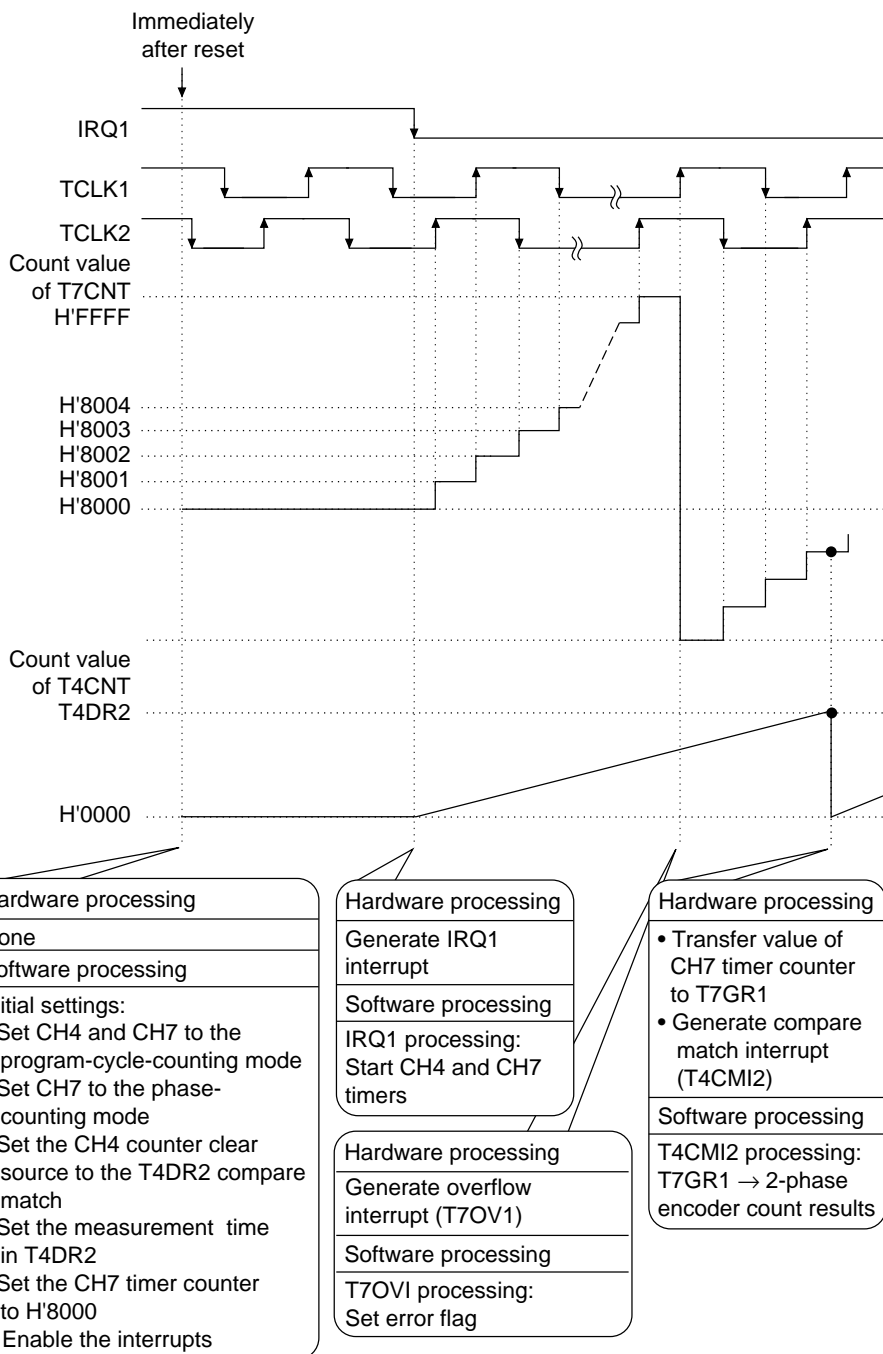
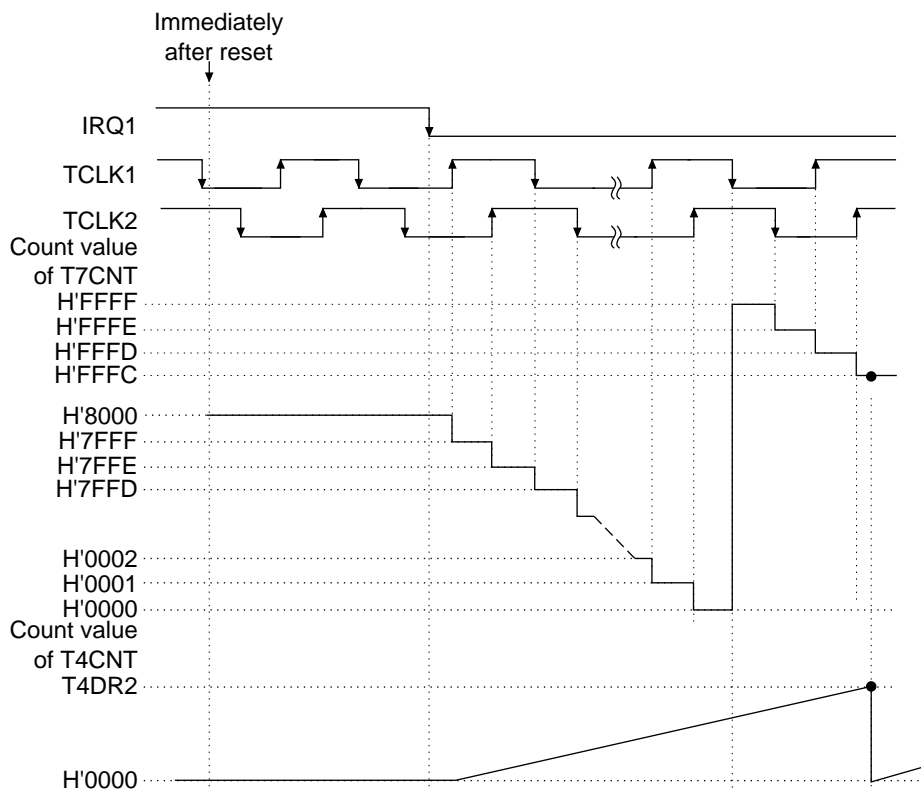


Figure 2.40 Operation of 2-Phase Encoder Count (Upcount)



Hardware processing

None

Software processing

Initial settings:

- Set CH4 and CH7 to the program-cycle-counting mode
- Set CH7 to the phase-counting mode
- Set the CH4 counter clear source to the T4DR2 compare match
- Set the measurement time in T4DR2
- Set the CH7 timer counter to H'8000
- Enable the interrupts

Hardware processing

Generate IRQ1 interrupt

Software processing

IRQ1 processing:
Start CH4 and CH7 timers

Hardware processing

Generate underflow interrupt (T7OVI)

Software processing

T7OVI processing:
Set error flag

Hardware processing

- Transfer value of CH7 timer counter to T7GR1
- Generate T4DR2 compare match interrupt (T4CMI2)

Software processing

T4CMI2 processing:
T7GR1 → 2-phase encoder count results

Figure 2.41 Operation of 2-Phase Encoder Count (Downcount)

2.8.4 Software

Tables 2.36 through 2.39 list software information for this function.

Table 2.36 Modules

Module Name	Label Name	Function
Main routine	ENPWMMN	Initializes 2-phase encoder count and 7-phase PWM output.
IRQ1 interrupt	CNTSTART	Starts the 2-phase encoder count.
Compare match interrupt	RAMSET	Sets the results of the 2-phase encoder count in RAM.
Overflow/underflow interrupt	FLGSET	Sets the error flag.

Table 2.37 Arguments

Label Name	Function	Data Length	Name of Module Used	I/O
CNT_RSLT	Sets the results of 2-phase encoder counting during the measurement time.	2 bytes	Main routine	O
ERRFLG	Indicates whether an overflow or underflow has occurred. 1: Occurred; 0: None.	1 bit	Overflow/underflow interrupt	O
CNT_TIM	Sets the measurement time.	2 bytes	Compare match interrupt	O
PWM_HI1 — PWM_H17	Sets the timer counter value corresponding to High width of the pulse. The pulse High width is found by the following equation. Pulse High width (ns) = Timer counter value × System clock cycle (100 ns when operating at 10 MHz) × Input clock division ratio for each channel.	2 bytes	Main routine	I
PWM_CYC	Sets the timer value corresponding to PWM cycle. The PWM cycle is found by the following equation. PWM cycle (ns) = Timer counter value × System clock cycle (100 ns when operating at 10 MHz) × Input clock division ratio for each channel.	2 bytes	Main routine	I

Table 2.38 Internal Registers Used

Register Name	Function	Name of Module Used
T4CRL	Selects ch4 counter clear source.	Main routine
T4DR2	Sets the measurement time.	Main routine
T7GR1	Sets the ch7 timer counter value at T4DR2 compare match.	Compare match interrupt
T1CRL—T3CRL	Selects timer counter clear sources. ch1 (master): DR2 compare match (selects synchronized clear source); ch2—ch3 (slave): Synchronized clear	Main routine
TMDRA	Synchronizes operation of ch1—ch3, and sets ch4 and ch7 to cycle-counting mode.	
TMDRB	Makes ch1—ch3 operate as PWM timers, and sets ch7 to phase-counting mode.	
T1DR2	Sets the PWM cycle.	
T1GR1	Sets the timer counter value that makes T1IOC1 output high.	
T1GR3	Sets the timer counter value that makes T1IOC1 output low.	
T1GR2	Sets the timer counter value that makes T1IOC2 output high.	
T1GR4	Sets the timer counter value that makes T1IOC2 output low.	
T1DR1	Sets the timer counter value that makes T1IOC1 output high.	
T1DR3	Sets the timer counter value that makes T1IOC1 output low.	
T2GR1	Sets the timer counter value that makes T2IOC1 output high.	
T2DR1	Sets the timer counter value that makes T2IOC1 output low.	
T2GR2	Sets the timer counter value that makes T2IOC2 output high.	
T2DR2	Sets the timer counter value that makes T2IOC2 output low.	
T3GR1	Sets the timer counter value that makes T3IOC1 output high.	
T3DR1	Sets the timer counter value that makes T3IOC1 output low.	
T3GR2	Sets the timer counter value that makes T3IOC2 output high.	
T3DR2	Sets the timer counter value that makes T3IOC2 output low.	
TSTR	Starts and stops ch1—ch7 as timer counters.	Main routine, compare match interrupt

Table 2.39 General Registers Used

Name of Module Used	Name of Register	Function
Main routine, compare match interrupt, overflow/underflow interrupt	R0	Used as work space when setting data.

RAM:

This task example does not use RAM except for arguments.

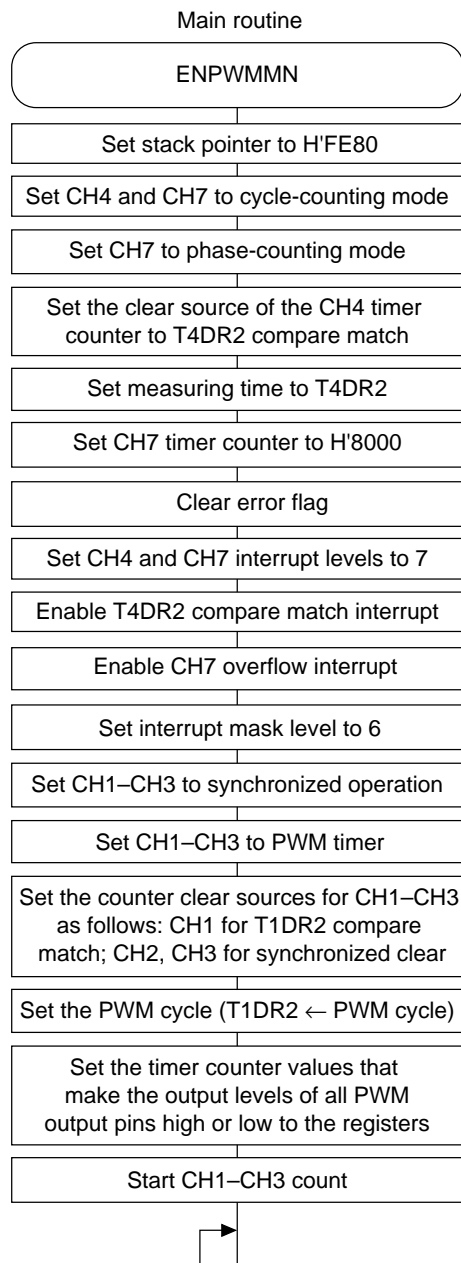
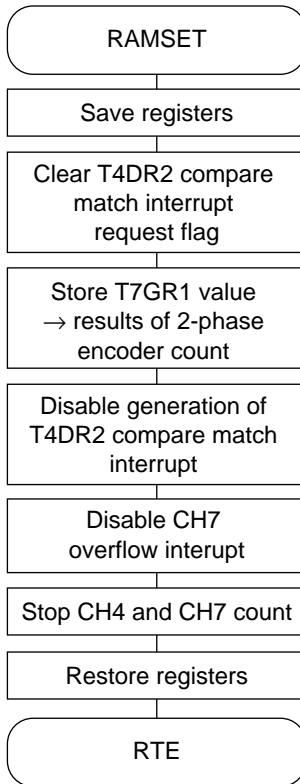
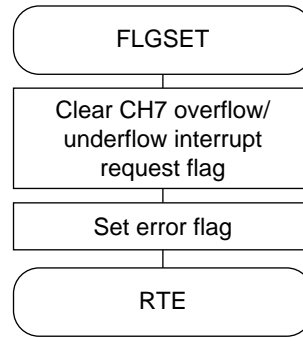


Figure 2.42 2-Phase Encoder Count and 7-Phase PWM Output (1 of 2)

2. Compare match interrupt



3. Overflow underflow interrupt



4. IRQ1 interrupt

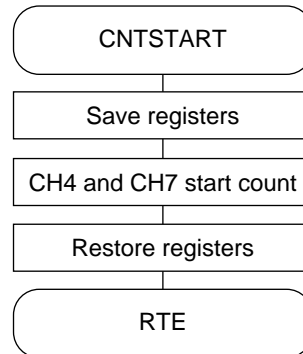


Figure 2.43 2-Phase Encoder Count and 7-Phase PWM Output (2 of 2)

2.9 A/D Converter Startup by IPU and A/D Conversion

Functions used: IPU, A/D.

2.9.1 Specifications

As shown in figure 2.44, a 12-channel voltage (0—3.5 V) is input to the H8/538 and the results of A/D conversion are stored in RAM.

The A/D converter is started up upon the T1DR2 compare match of the IPU.

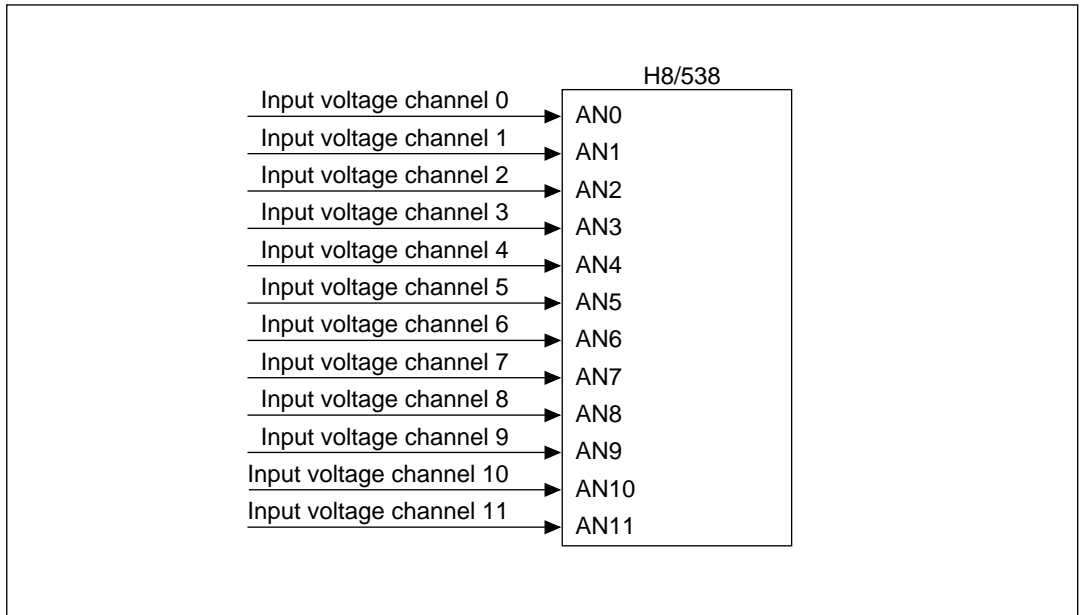


Figure 2.44 Block Diagram of Voltage Measurement Using the H8/538

2.9.2 Functions Used

This task example uses the following functions of the A/D converter:

- Function for automatically doing A/D conversion of channels 1—12 (AN0—AN11) without software (scan mode)
- Function for starting up the A/D converter on IPU compare match
- Function for generating interrupt at end of A/D conversion

Figure 2.45 is a block diagram of A/D converter startup and 12-channel A/D conversion.

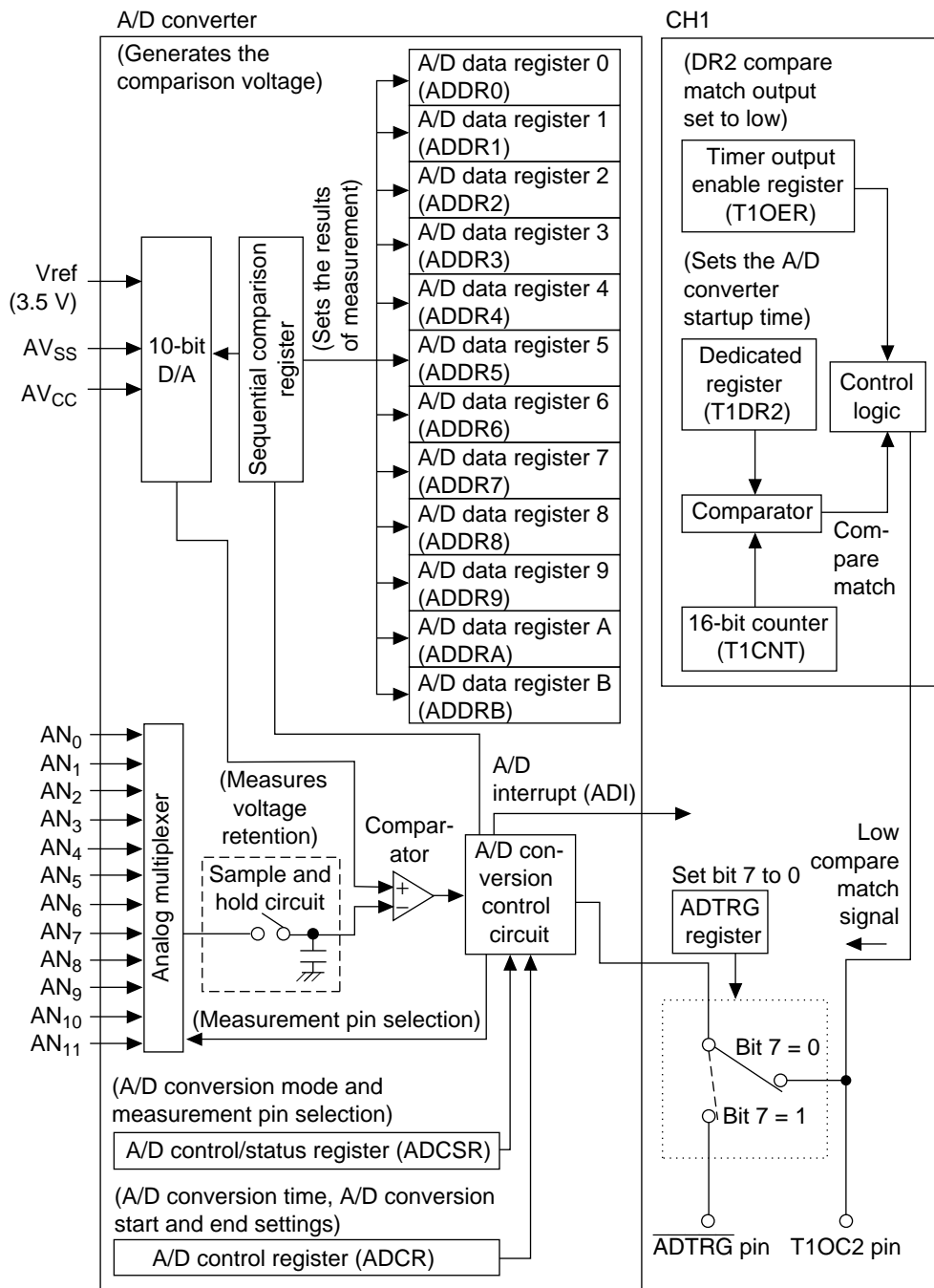


Figure 2.45 Block Diagram of A/D Converter Startup and A/D Conversion

Table 2.40 lists the function allocations for this task example.

Table 2.40 A/D Converter and IPU Function Allocation

Register Name	Function
ADCSR	Selects the A/D conversion mode (simple mode/scan mode) and the measurement pin.
ADCR	Sets the conversion time and the start/stop of measurement.
ADDR0—ADDRB	Stores the results of A/D conversion.
T1DR2	Sets the startup period of the A/D converter.

2.9.3 Operation

Figure 2.47 shows the principles of operation. As the figure shows, T1DR2 compare match starts up the A/D converter and the voltage input to AN0—AN11 is A/D sequentially converted three times in groups of 4 channels, AN0—AN3, AN4—AN7, and AN8—AN11. The results of A/D conversion stored in ADDR0—ADDRB are stored in the 24 bytes of RAM SCN_RE0—SCN_RE11.

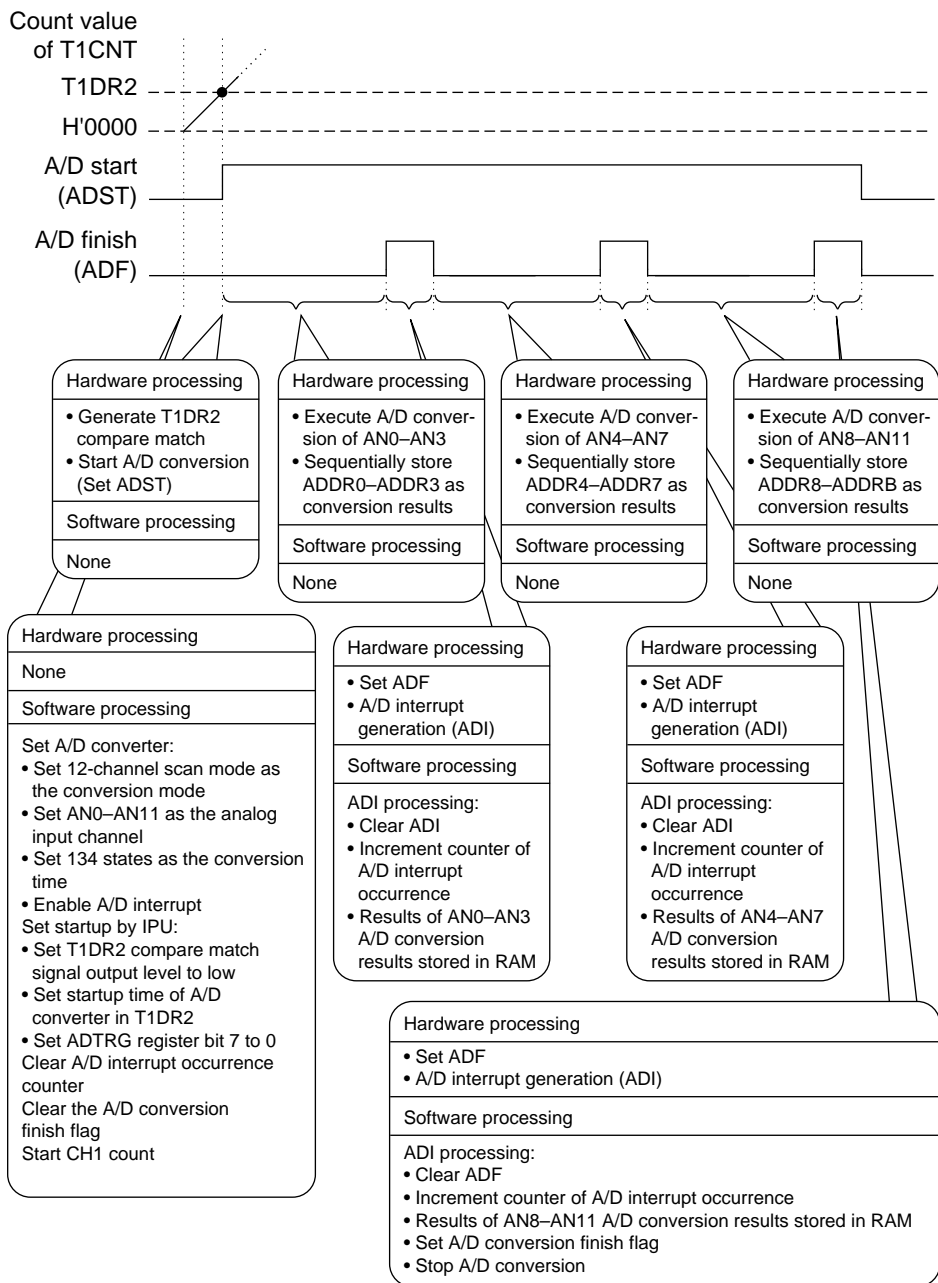


Figure 2.46 Principles of A/D Converter Operation

2.9.4 Software

Tables 2.41 through 2.45 list software information for this function.

Table 2.41 Modules

Module Name	Label Name	Function
Main routine	ADSCNMN	Sets the A/D converter and A/D converter startup by IPU.
A/D interrupt	SCNEND	Upon ADI, stores A/D conversion results to RAM and stops A/D conversion.

Table 2.42 Arguments

Label Name, Register Name	Function	Data Length	Name of Module Used	I/O
SCN_RE0	Sets the results of 12-channel A/D conversion. The 10-bit conversion results are set as follows.	2 bytes	A/D interrupt	O
<div><div>SCN_RE0 to SCN_RE7 upper byte</div><div>SCN_RE0 to SCN_RE7 lower byte</div><div><div>Bit 7</div><div>AD9AD8AD7AD6AD5AD4AD3AD2</div><div>AD1AD0</div><div>Bit 0</div></div><div>AD0–AD9 indicates the bit number of the A/D conversion results</div></div>				
SCN_RE1				
SCN_RE2				
SCN_RE3				
SCN_RE4				
SCN_RE5				
SCN_RE6				
SCN_RE7				
SCN_RE8				
SCN_RE9				
SCN_RE10				
SCN_RE11				
SCN_ENDF	Flag indicating whether the 12-channel A/D conversion is completed.	1 bit	A/D interrupt Main routine	O I
ADSTRTIM	Sets the counter value at startup of the A/D converter.	2 bytes	Main routine	I

Table 2.43 Internal Registers Used

Register Name	Function	Name of Module Used
ADCSR	Selects A/D conversion mode (simple mode/scan mode), analog input channel and enable/disable A/D interrupt at end of A/D conversion.	Main routine A/D interrupt
ADCR	Selects A/D conversion time.	Main routine
ADDR0— ADDRB	Stores A/D conversion results.	A/D interrupt
T1OERA	Selects ch1 compare match output level.	Main routine
T1DR2	Output compare register that sets A/D converter startup time.	Main routine
ADTRG	Selects enable/disable of A/D converter startup by IPU.	Main routine
TSTR	Starts and stops ch1—ch7 timer counters.	Main routine

Table 2.44 General Registers Used

Name of Module Used	Name of Register	Function
A/D interrupt	R0	Used as work space when storing data.

Table 2.45 RAM

Name of Module Used	Name of Label	Function
A/D interrupt	ADICNT	Counts the number of A/D interrupts generated.

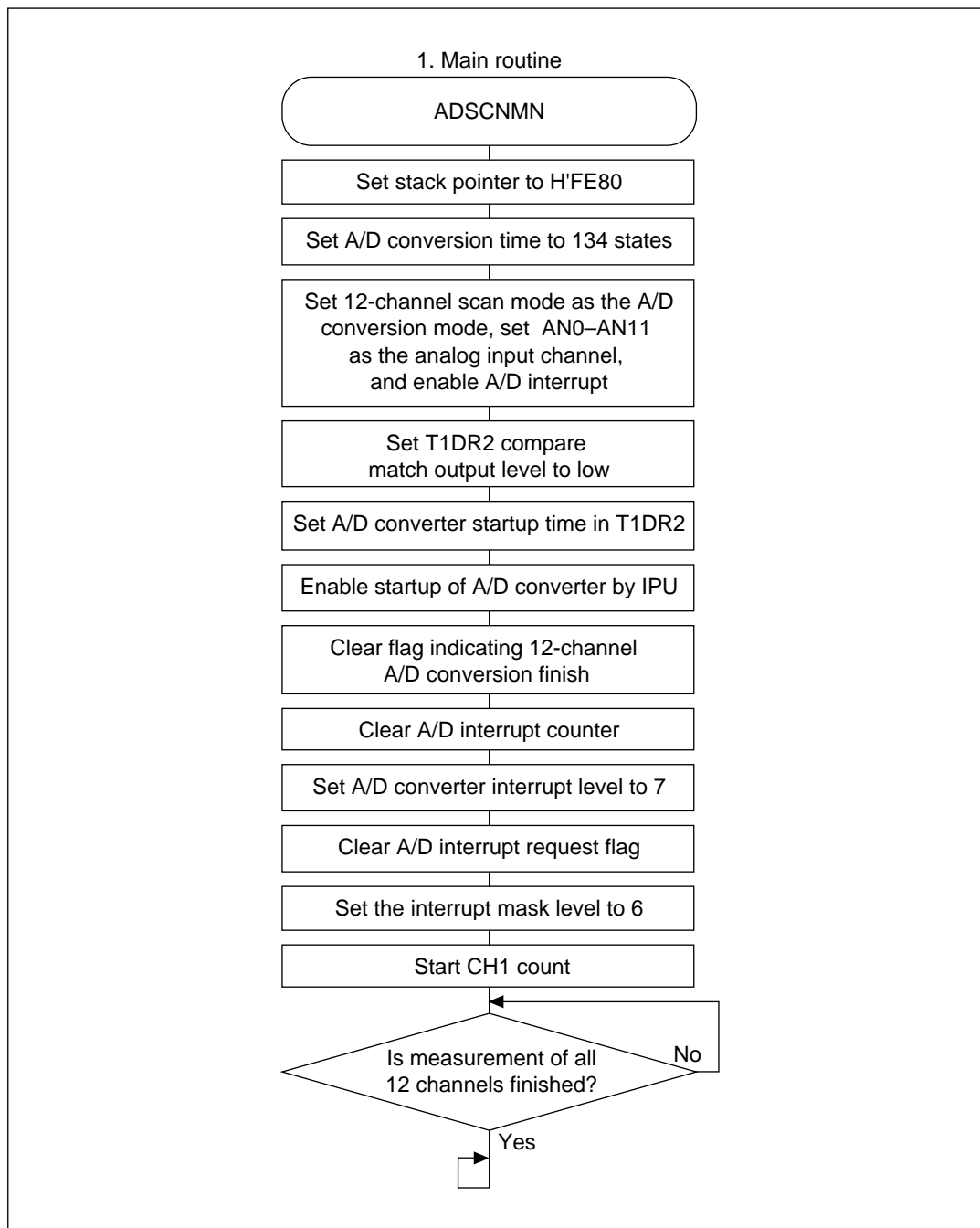


Figure 2.47 A/D Conversion (1 of 2) Flowchart

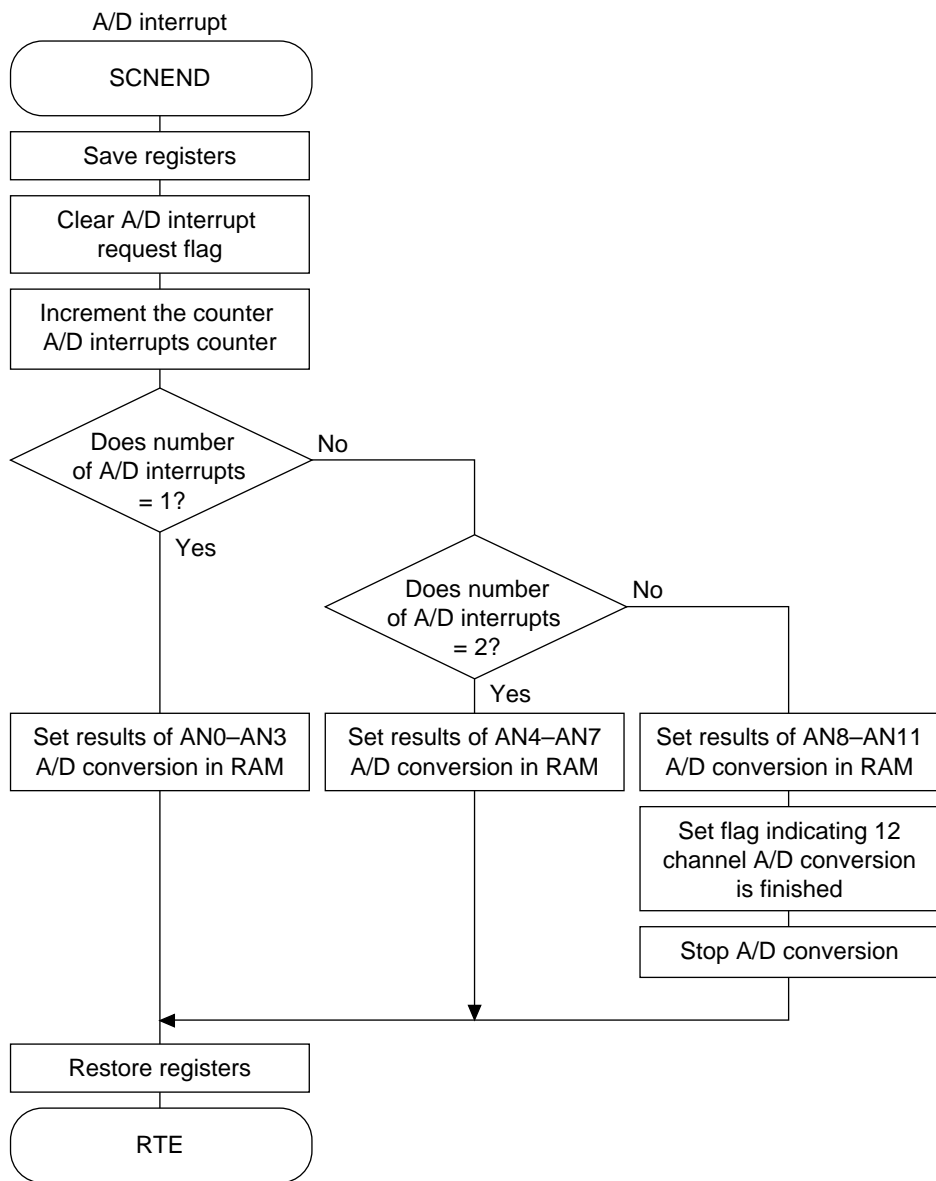


Figure 2.48 A/D Conversion (2 of 2) Flowchart

2.10 Multiprocessor Communication

Functions used: SCI (Multiprocessor communication).

2.10.1 Specifications

1. As shown in Figure 2.49, three H8/538 units send and receive data using a common serial communications line.
2. The master H8/538 transfers 1 byte of data to each slave H8/538 and the slave H8/538 units receive only the data sent to them individually.
3. The data is sent and received at 9600 bps, 8-bit data, 1 stop bit and no-parity.

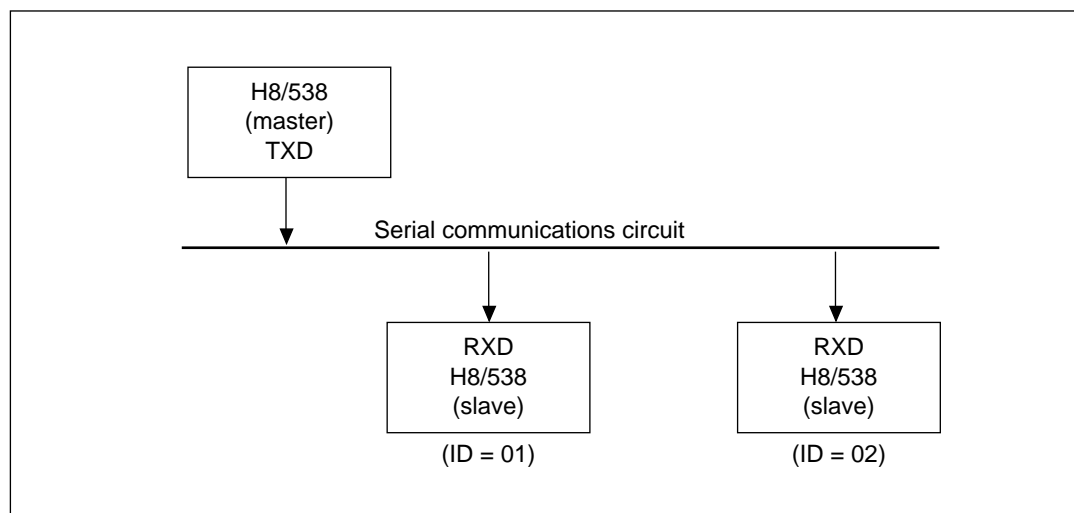


Figure 2.49 Asynchronous SCI Interface that Uses Multiprocessor Functions

2.10.2 Functions Used

This task example uses the multiprocessor communication functions of the SCI to perform multiprocessor communications.

- Figure 2.50 is a block diagram of the master SCI used in this task example. The SCI shown in the block diagram uses the following functions for master side transmission:
- Function for communicating in a format with an appended multiprocessor bit (multiprocessor communication function).
- Function for generating an interrupt at the start of transmission (TXI interrupt).
- Function for generating an interrupt at the end of transmission (TEI interrupt).

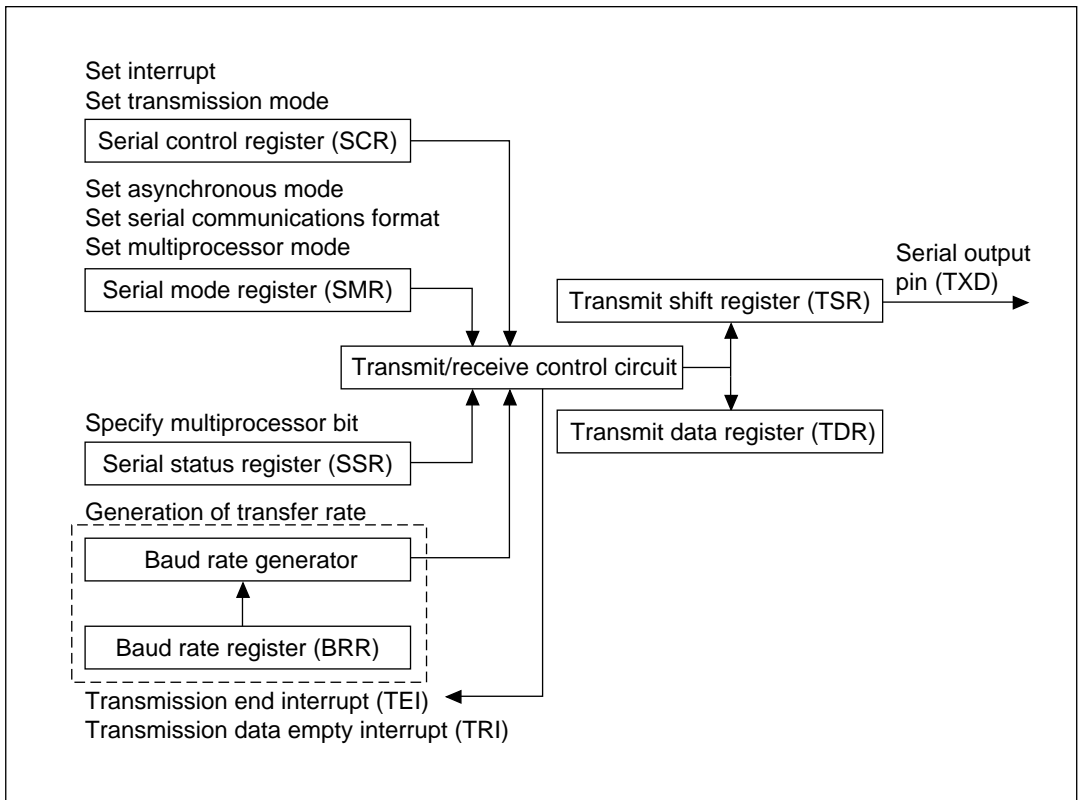


Figure 2.50 Master SCI Block Diagram

Figure 2.51 is a block diagram of the slave SCI used in this task example. Table 2.46 lists SCI function allocation. SCI functions are allocated as shown in the table to perform multiprocessor communications. The SCI shown in the block diagram uses the following functions for slave side reception:

- Function for asynchronous data communication, which synchronizes in character units (asynchronous mode)
- Function for communicating in a format with an appended multiprocessor bit (multiprocessor communication function)
- Function for generating an interrupt when a multiprocessor bit is received (multiprocessor interrupt)
- Function for generating an interrupt at the end of reception (RXI interrupt)

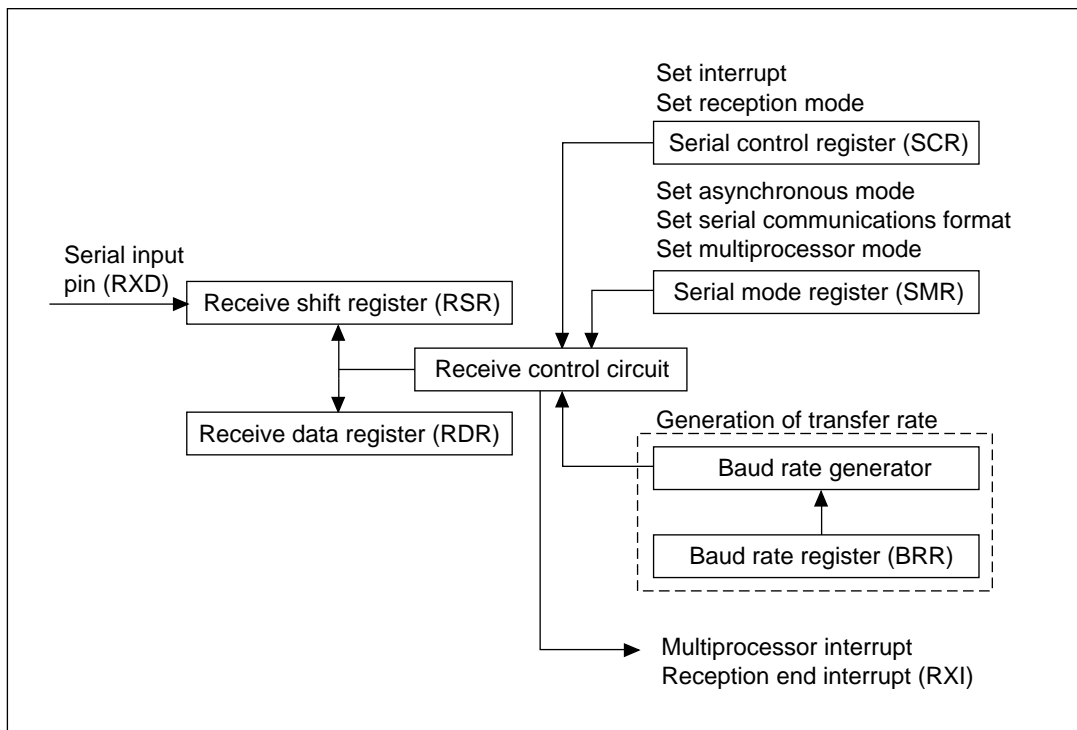


Figure 2.51 Slave SCI Block Diagram

Table 2.46 SCI Function Allocation

SCI Function	Function
RXD	Receives data from H8/538.
TXD	Transmits data to H8/538.
SMR	Sets SCI to asynchronous mode and multiprocessor mode.
SCR	Enables transmit/receive interrupts and sets SCI to transmit/receive mode.
SSR	Sets start-transmission/multiprocessor bit.
RDR	Sets data received from H8/538.
TDR	Sets data to transmit to H8/538.
BRR	Sets transfer speed.

2.10.3 Operation

Figures 2.52 through 2.54 show slave side operation.

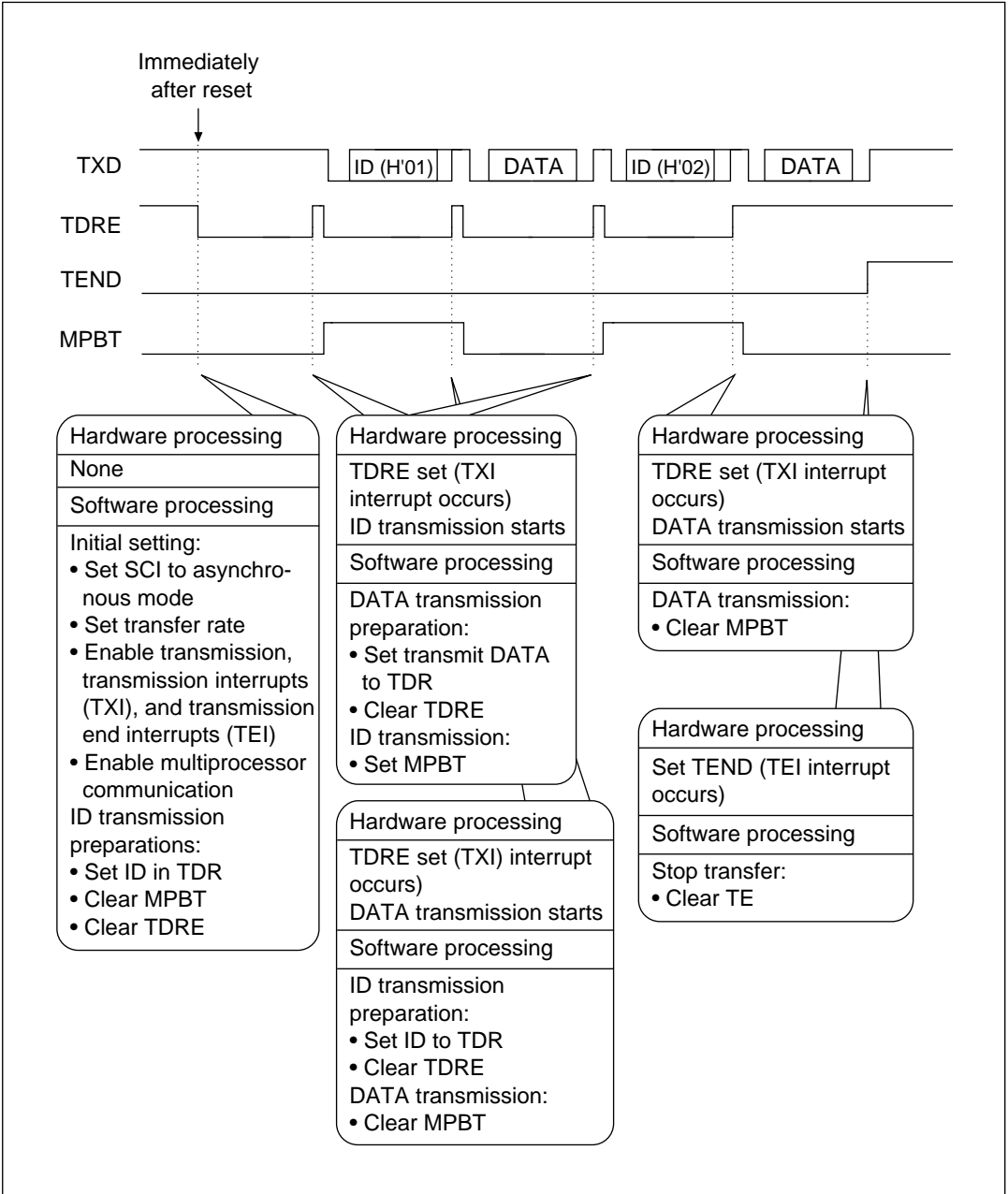


Figure 2.52 Multiprocessor Communication (Master Side)

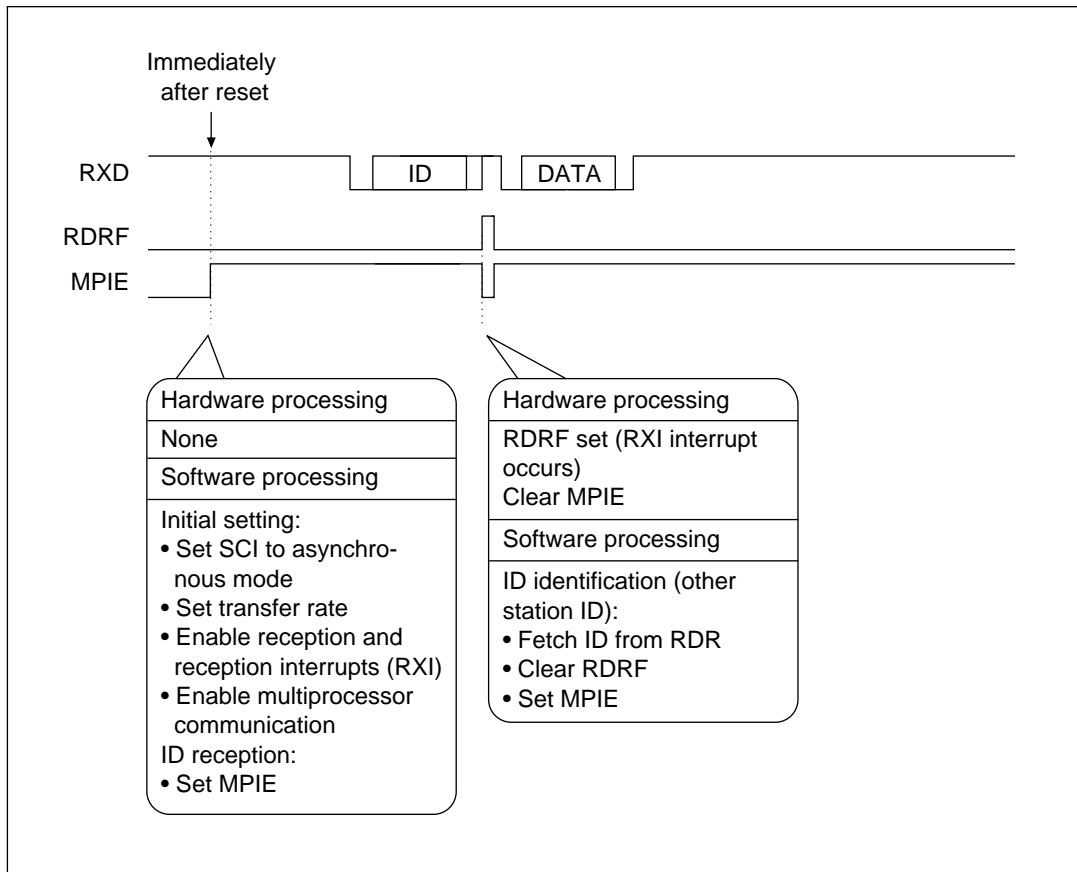


Figure 2.53 Slave Side of SCI (Operation When the ID of Another Station is Received)

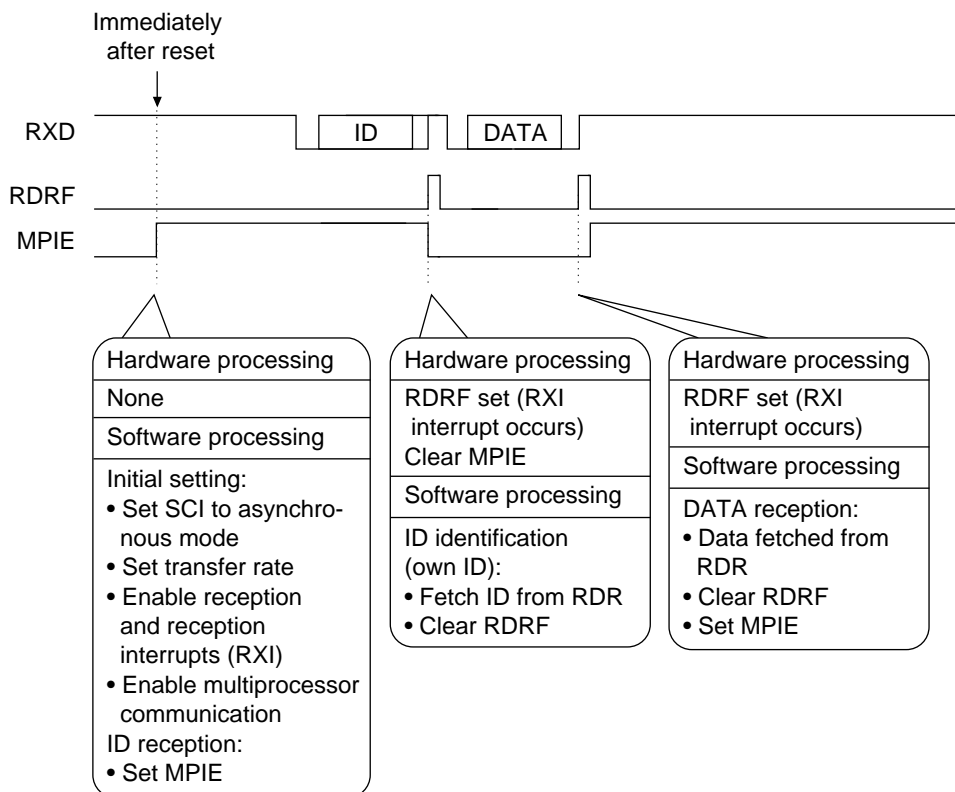


Figure 2.54 Block Diagram of Slave Side of SCI (Operation when Own ID is Received)

2.10.4 Software

Tables 2.47 through 2.50 list software information for this function.

Master:

Table 2.47 Modules

Module Name	Label Name	Function
Main routine	MPMASMN	Initializes SCI.
Data transmission	MPSCITX	Starts up on TXI interrupt and transmits ID and data.
Transmission end	MPSCITE	Starts up on TEI interrupt and stops transmission.

Table 2.48 Arguments

Label Name, Register Name	Function	Data Length	Name of Module Used	I/O
TXDATA	Pointer to transmit data (ID, data)	2 bytes	Main routine	O
			Data transmission	I

Table 2.49 Internal Registers Used

Register Name	Function	Name of Module Used
SMR	Sets communication mode, send/receive format and selection of clock to baud rate generator.	Main routine
SCR	Selects enable/disable for interrupts, enable/disable for send/receive operation, and SCI clock source.	Main routine Transmission end
SSR	Sets the status flag indicating the SCI operating status and the transmitting multiprocessor bit and stores the multiprocessor bit received.	Main routine Data transmission
TDR	Sets data to be transmitted.	Main routine Data transmission
BRR	Sets send/receive bit rate.	Main routine

Table 2.50 General Registers Used

Name of Module Used	Name of Register	Function
Main routine	R0, R1	Used as work space when sending transmit data to TDR.
Data transmission		

RAM:

This task example does not use RAM except for arguments.

Slave:

Tables 2.51 through 2.55 list software information for this function.

Table 2.51 Modules

Module Name	Label Name	Function
Main routine	MPSRVMN	Initializes SCI.
Data reception	MPSCIRX	Starts up on RXI interrupt and receives data addressed to itself only.

Table 2.52 Arguments

Label Name, Register Name	Function	Data Length	Name of Module Used	I/O
R0	Sets received data.	1 byte	Data reception	I
			Main routine	O
DTRCVF	Flag indicating data reception. 1: Data received; 0: No data received.	1 bit	Data reception	O
			Main routine	I

Table 2.53 Internal Registers Used

Register Name	Function	Name of Module Used
SMR	Sets communication mode, send/receive format and selection of clock to baud rate generator.	Main routine
SCR	Selects enable/disable for interrupts, enable/disable for send/receive operation, and SCI clock source.	Main routine Data reception
SSR	Sets the status flag indicating the SCI operating status and the transmitting multiprocessor bit and stores the multiprocessor bit received.	Data reception
RDR	Stores data received.	Data reception
BRR	Sets send/receive bit rate.	Main routine
IPRF	Sets the SCI interrupt priority order.	Main routine

Table 2.54 General Registers Used

Name of module	Name of Register	Function
Main routine	R0	Used as argument that transfers received data to main routine.
Data reception		

Table 2.55 RAM Used

Name of Label	Name of Module Used	Data Length	Function
RXDATA	Main routine	1 byte	Stores received data.

2.10.5 Flowcharts

Master Operation:

Figure 2.55 shows the principles of operation of this task on the master side. Hardware and software run on the timing shown in the figure to send data to the receiving H8/538 units.

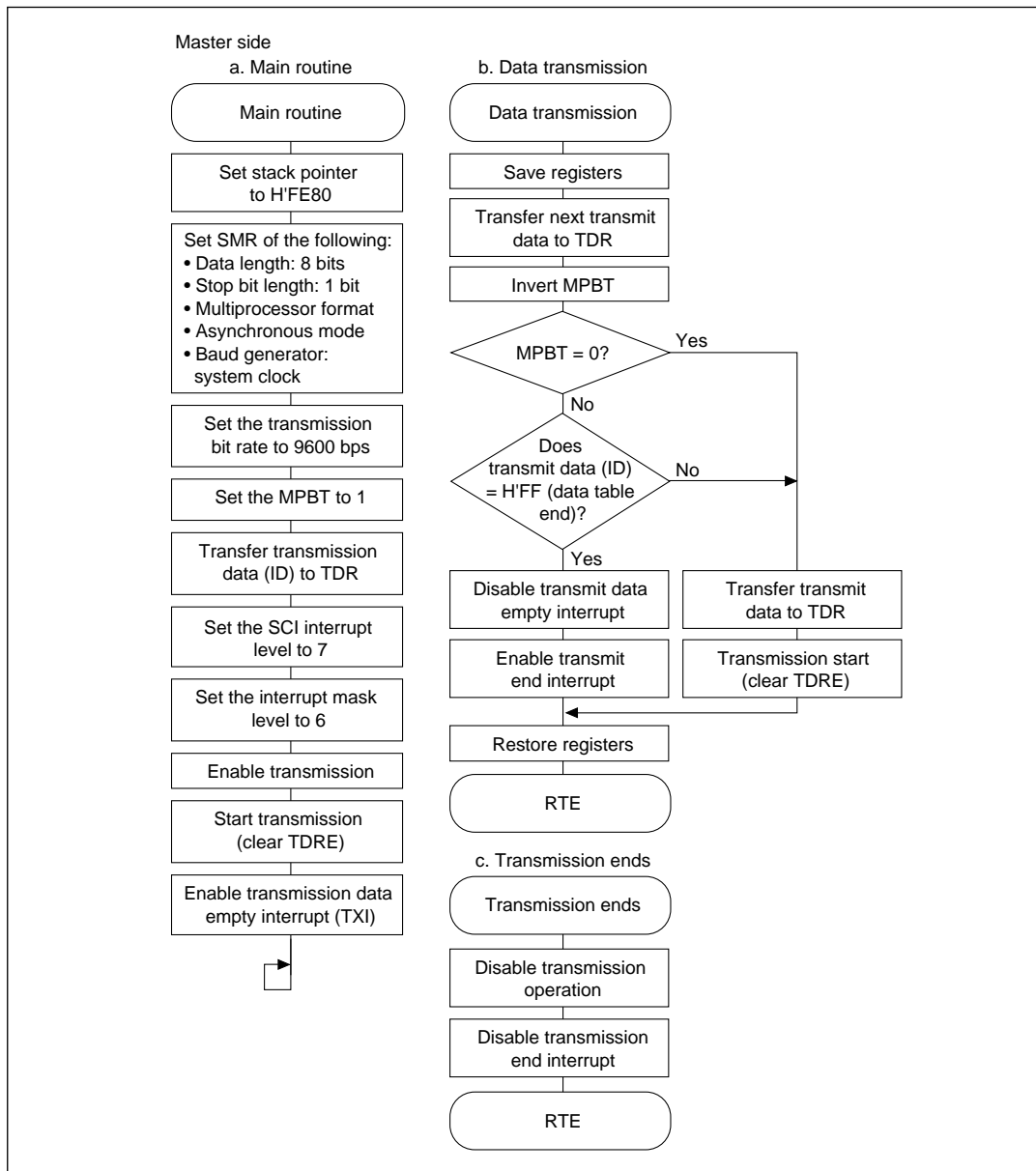


Figure 2.55 Multiprocessor Communication Operation (Master Side)

Slave Operation:

Figure 2.56 shows the principles of operation of this task on the slave side. Hardware and software run on the timing shown in the figure to receive data from the transmitting H8/538 unit.

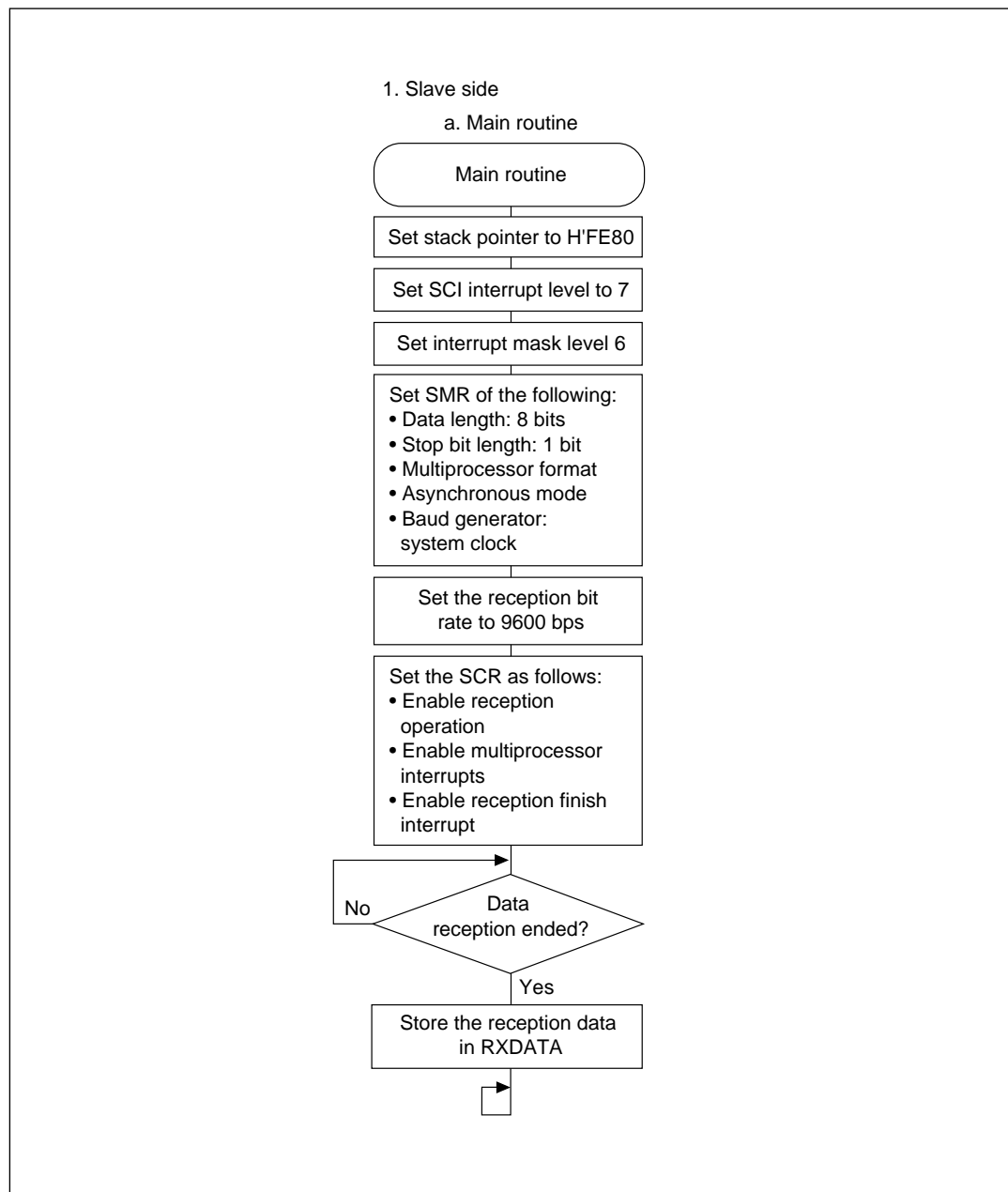


Figure 2.56 Multiprocessor Communication Operation (Slave Side)

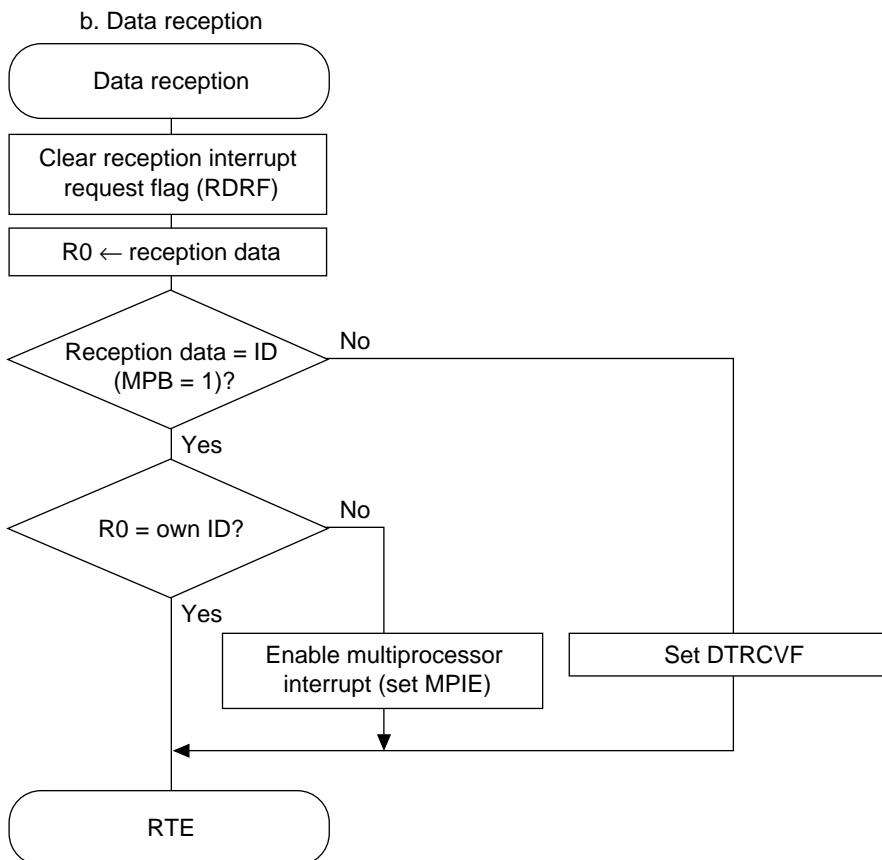


Figure 2.57 Multiprocessor Communication Flowchart