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.

4283 Group

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

REJ03B0109-0101
Rev.1.01
2006.03.20

DESCRIPTION

The 4283 Group enables fabrication of 8 × 7 key matrix and has the followin timers;

- an 8-bit timer which can be used to set each carrier wave and has two reload register
- an 8-bit timer which can be used to auto-control and has a reload register.

FEATURES

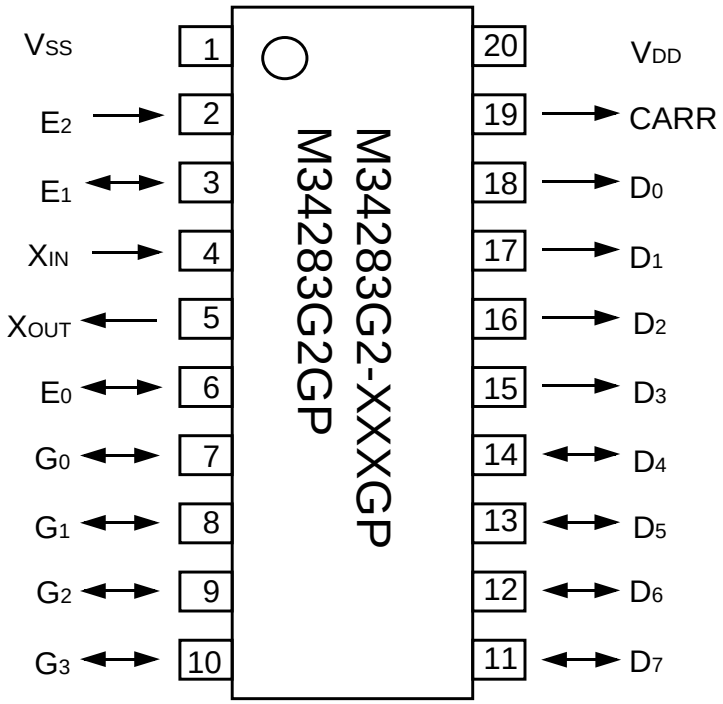
- Number of basic instructions 68
- Minimum instruction execution time ..... 8.0 μs (at f(X_{IN}) = 4.0 MHz, system clock = f(X_{IN})/8)
- Supply voltage ..... 1.8 V to 3.6 V
- Subroutine nesting ..... 4 levels
- Timer
  Timer 1 ..... 8-bit timer (This has a reload register and carrier wave output auto-control function)
  Timer 2 ..... 8-bit timer (This has two reload registers and carrier wave output function)
- Logic operation function (XOR, OR, AND)
- RAM back-up function
- Key-on wakeup function (ports D_4–D_7, E_0–E_2, G_0–G_3) 11
- I/O port (ports D, E, G, CARR) 16
- Oscillation circuit ..... Ceramic resonance
- Watchdog timer
- Power-on reset circuit
- Voltage drop detection circuit ..... Typical:1.50 V (system reset)

APPLICATION

Various remote control transmitters

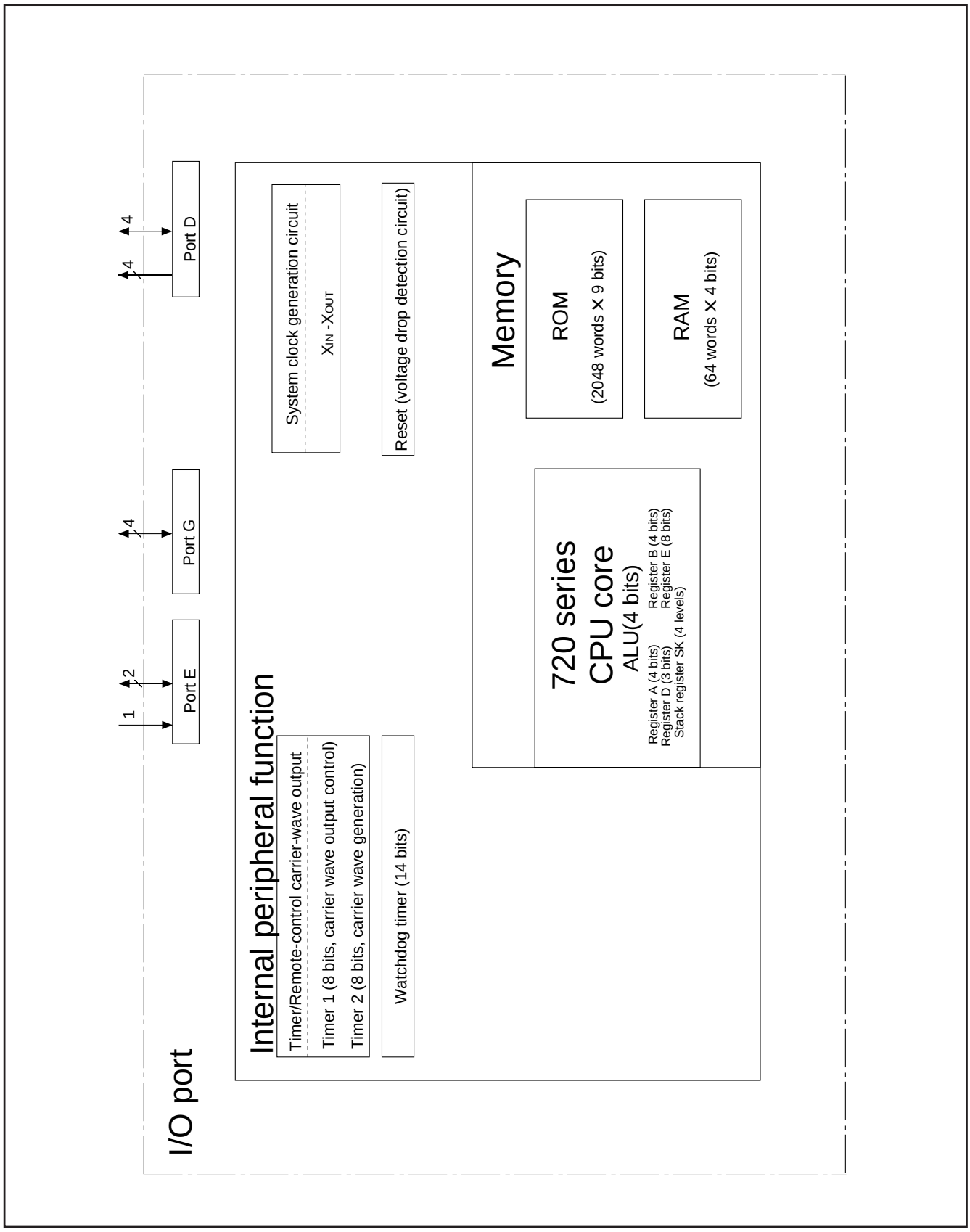
Part number	ROM size (× 9 bits)	RAM size (× 4 bits)	Package	ROM type
M34283G2-XXXGP	2048 words	64 words	20P2F-A	QzROM
M34283G2GP	2048 words	64 words	20P2F-A	QzROM (blank)

PIN CONFIGURATION (TOP VIEW)



Outline 20P2F-A

BLOCK DIAGRAM



PERFORMANCE OVERVIEW

Parameter			Function
Number of basic instructions			68
Minimum instruction execution time			8.0 μs (f(X_{IN}) = 4.0 MHz, system clock = f(X_{IN})/8, V_{DD} = 3 V)
Memory sizes	ROM		2048 words × 9 bits
	RAM		64 words × 4 bits
Input/Output ports	D_0–D_3	Output	Four independent output ports
	D_4–D_7	I/O	Four independent I/O ports with the pull-down function
	E_0–E_2	Input	3-bit input port with the pull-down function
	E_0, E_1	Output	2-bit output port (E_0, E_1)
	G_0–G_3	I/O	4-bit I/O port with the pull-down function
	CARR	Output	1-bit output port; CMOS output
Timer	Timer 1		8-bit timer with a reload register
	Timer 2		8-bit timer with two reload registers
Subroutine nesting			4 levels (However, only 3 levels can be used when the TABP p instruction is executed)
Device structure			CMOS silicon gate
Package			20-pin plastic molded SSOP (20P2F-A)
Operating temperature range			–20 °C to 85 °C
Supply voltage			1.8 V to 3.6 V
Power dissipation (typical value)	Active mode		400 µA
			(f(X_{IN}) = 4.0 MHz, system clock = f(X_{IN})/8, V_{DD} = 3 V)
	RAM back-up mode		0.1 μA (Ta=25°C, V_{DD} = 3 V)

PIN DESCRIPTION

Pin	Name	Input/Output	Function
V_{DD}	Power supply	—	Connected to a plus power supply.
V_{SS}	Ground	—	Connected to a 0 V power supply.
X_{IN}	System clock input	Input	I/O pins of the system clock generating circuit. Connect a ceramic resonator between pins X_{IN} and X_{OUT}. The feedback resistor is built-in between pins X_{IN} and X_{OUT}.
X_{OUT}	System clock output	Output	
D_0–D_3	Output port D	Output	Each pin of port D has an independent 1-bit wide output function. The output structure is P-channel open-drain.
D_4–D_7	I/O port D	I/O	1-bit I/O port. For input use, set the latch of the specified bit to “0.” When the built-in pull-down transistor is turned on, the key-on wakeup function using “H” level sense and the pull-down transistor become valid. The output structure is P-channel open-drain.
E_0–E_2	I/O port E	Output	2-bit (E_0, E_1) output port. The output structure is P-channel open-drain.
		Input	3-bit input port. For input use (E_0, E_1), set the latch of the specified bit to “0.” When the built-in pull-down transistor is turned on, the key-on wakeup function using “H” level sense and the pull-down transistor become valid. Port E_2 has an input-only port and has a key-on wakeup function using “H” level sense and pull-down transistor.
G_0–G_3	I/O port G	I/O	4-bit I/O port. For input use, set the latch of the specified bit to “0.” The output structure is P-channel open-drain. When the built-in pull-down transistor is turned on, the key-on wakeup function using “H” level sense and pull-down transistor become valid.
CARR	Carrier wave output for remote control	Output	Carrier wave output pin for remote control. The output structure is CMOS circuit.

CONNECTIONS OF UNUSED PINS

Pin	Connection	Usage condition
D_0–D_3	Open.	
	Connect to V_{DD}.	
D_4–D_7	Open (Set the output latch to "1").	Pull-down transistor OFF.
	Open (Set the output latch to "0").	
	Connect to V_{DD}.	Pull-down transistor OFF.
E_0, E_1	Open (Set the output latch to "1").	Pull-down transistor OFF.
	Open (Set the output latch to "0").	
	Connect to V_{DD}.	Pull-down transistor OFF.
E_2	Open.	
	Connect to V_{SS}.	
G_0–G_3	Open (Set the output latch to "1").	Pull-down transistor OFF.
	Open (Set the output latch to "0").	
	Connect to V_{DD}.	Pull-down transistor OFF.
CARR	Open.	

(Note when connecting to V_{SS} and V_{DD})
- Connect the unused pins to V_{SS} or V_{DD} at the shortest distance and use the thick wire against noise.

PORT FUNCTION

Port	Pin	Input/ Output	Output structure	Control bits	Control instructions	Control registers	Remark
Port D	D_0–D_3	Output (4)	P-channel open-drain	1 bit	SD RD CLD		
	D_4–D_7	I/O (4)			SD RD CLD SZD	PU1	Pull-down function and key-on wakeup function (programmable)
Port E	E_0 E_1	I/O (2)	P-channel open-drain	Output: 2 bits Input: 3 bits	OEA IAE	PU0	Pull-down function and key-on wakeup function (programmable)
	E_2	Input (1)			IAE		
Port G	G_0–G_3	I/O (4)	P-channel open-drain	4 bits	OGA IAG	PU0	Pull-down function and key-on wakeup function (programmable)
Port CARR	CARR	Output (1)	CMOS	1 bit	SCAR RCAR		

DEFINITION OF CLOCK AND CYCLE

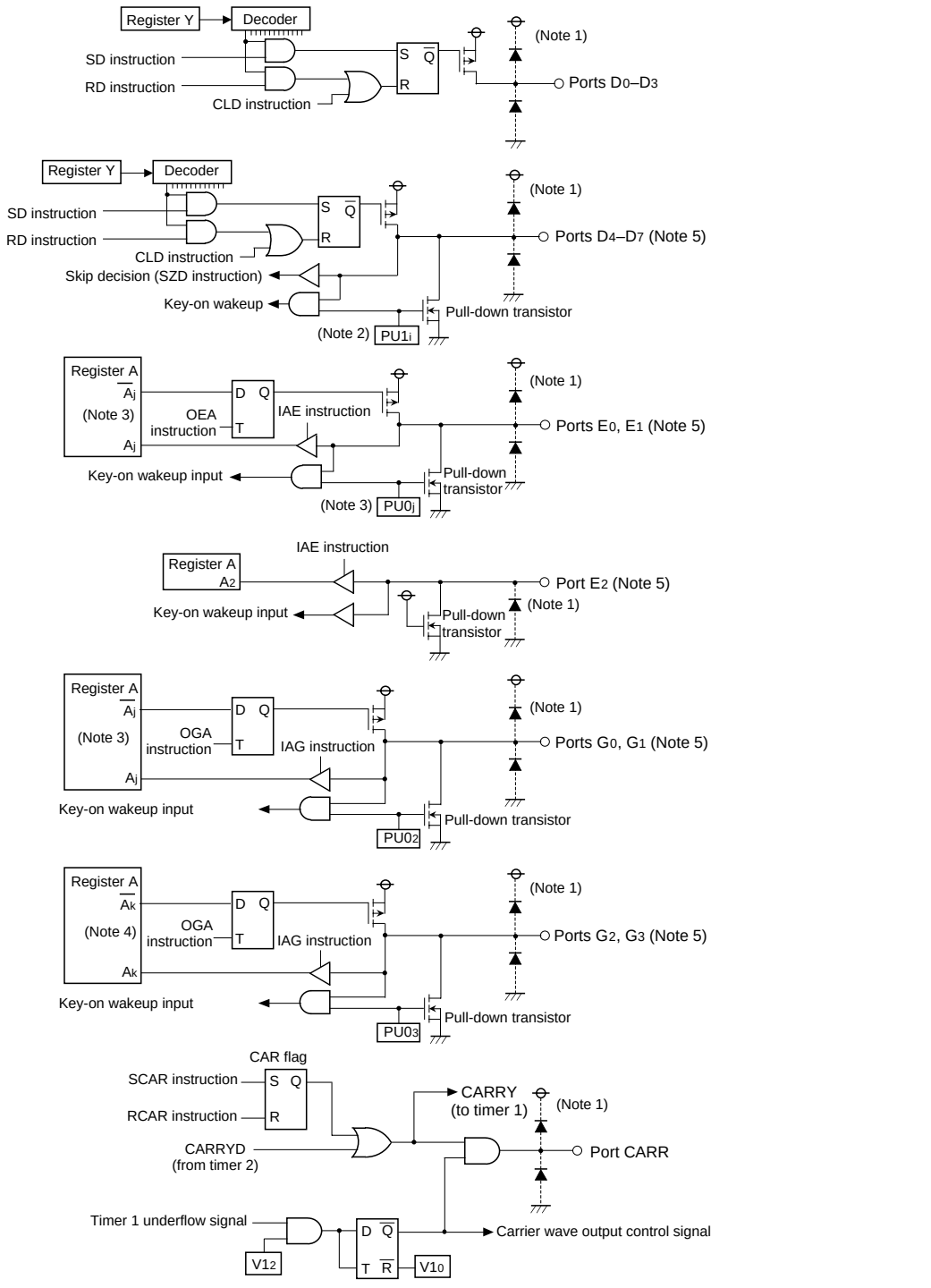
- System clock (STCK)
  The system clock is the source clock for controlling this product. It can be selected as shown below whether to use the CCK instruction.

CCK instruction	System clock	Instruction clock
When not using	f(X_{IN})/8	f(X_{IN})/32
When using	f(X_{IN})	f(X_{IN})/4

- Instruction clock (INSTCK)
  The instruction clock is a signal derived by dividing the system clock by 4, and is the basic clock for controlling CPU. The one instruction clock cycle is equivalent to one machine cycle.

- Machine cycle
  The machine cycle is the cycle required to execute the instruction.

PORT BLOCK DIAGRAMS



Notes 1: This symbol represents a parasitic diode.
2: i represents bits 0 to 3.
3: j represents bits 0, 1.
4: k represents bits 2, 3.
5: Applied voltage must be less than VDD.

FUNCTION BLOCK OPERATIONS
CPU

(1) Arithmetic logic unit (ALU)

The arithmetic logic unit ALU performs 4-bit arithmetic such as 4-bit data addition, comparison, and bit manipulation.

(2) Register A and carry flag

Register A is a 4-bit register used for arithmetic, transfer, exchange, and I/O operation.

Carry flag CY is a 1-bit flag that is set to “1” when there is a carry with the AMC instruction (Figure 1).

It is unchanged with both A n instruction and AM instruction.

The value of A_0 is stored in carry flag CY with the RAR instruction (Figure 2).

Carry flag CY can be set to “1” with the SC instruction and cleared to “0” with the RC instruction.

(3) Registers B and E

Register B is a 4-bit register used for temporary storage of 4-bit data, and for 8-bit data transfer together with register A.

Register E is an 8-bit register. It can be used for 8-bit data transfer with register B used as the high-order 4 bits and register A as the low-order 4 bits (Figure 3).

(4) Register D

Register D is a 3-bit register.

It is used to store a 7-bit ROM address together with register A and is used as a pointer within the specified page when the TABP p, BLA p, or BMLA p instruction is executed (Figure 4).

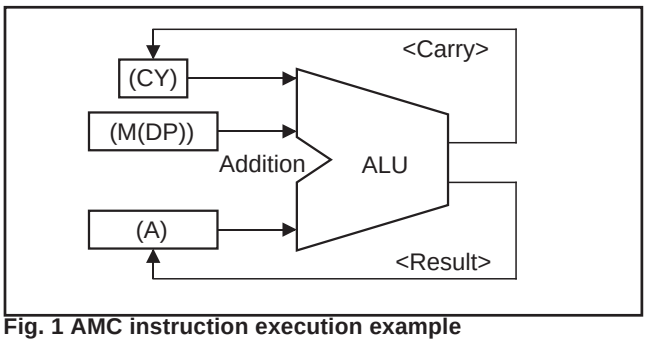


Fig. 1 AMC instruction execution example

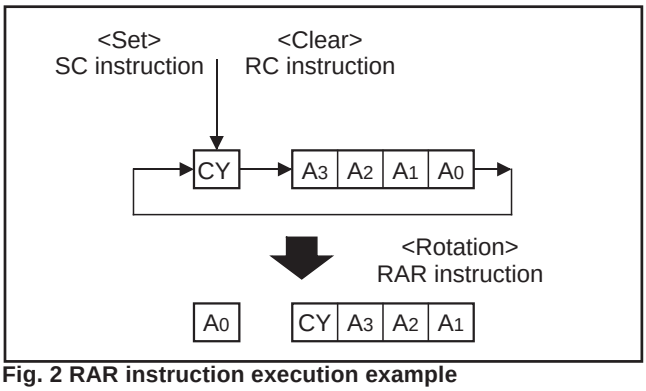


Fig. 2 RAR instruction execution example

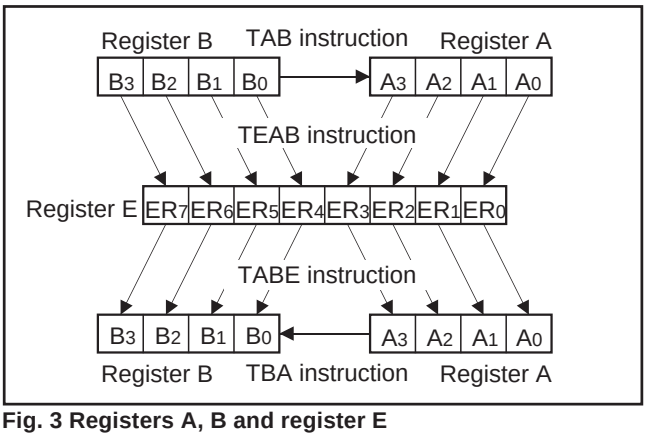


Fig. 3 Registers A, B and register E

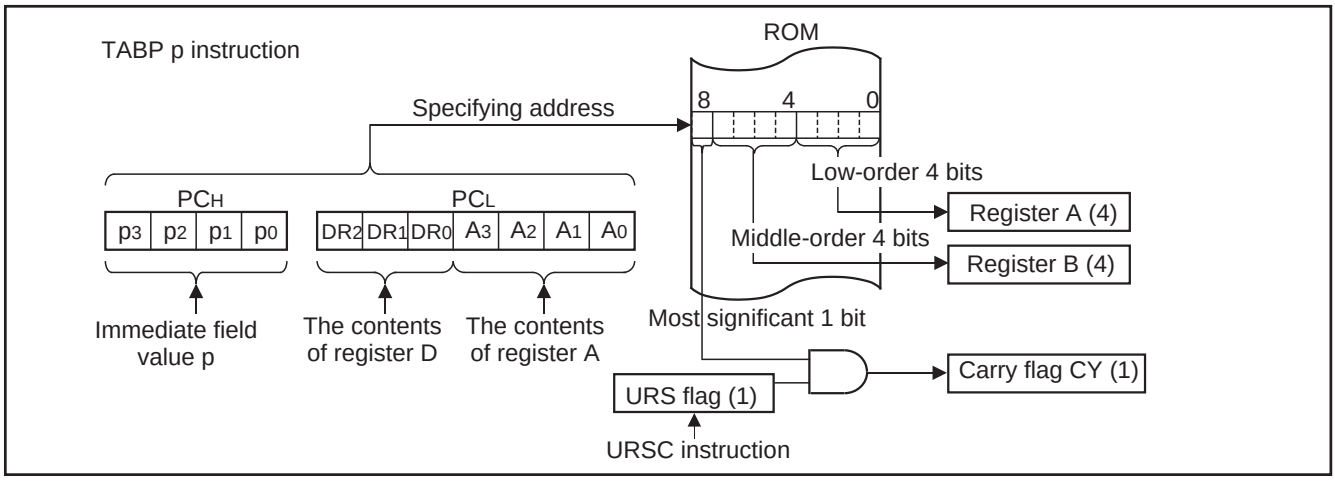


Fig. 4 TABP p instruction execution example

(5) Most significant ROM code reference enable flag (URS)

URS flag controls whether to refer to the contents of the most significant 1 bit (bit 8) of ROM code when executing the TABP p instruction. If URS flag is "0," the contents of the most significant 1 bit of ROM code is not referred even when executing the TABP p instruction. However, if URS flag is "1," the contents of the most significant 1 bit of ROM code is set to flag CY when executing the TABP p instruction (Figure 4).

URS flag is "0" after system is released from reset and returned from RAM back-up mode. It can be set to "1" with the URSC instruction, but cannot be cleared to "0."

(6) Stack registers (SKs) and stack pointer (SP)

Stack registers (SKs) are used to temporarily store the contents of program counter (PC) just before branching until returning to the original routine when;

- performing a subroutine call, or
- executing the table reference instruction (TABP p).

Stack registers (SKs) are four identical registers, so that subroutines can be nested up to 4 levels. However, one of stack registers is used when executing a table reference instruction. Accordingly, be careful not to over the stack. The contents of registers SKs are destroyed when 4 levels are exceeded.

The register SK nesting level is pointed automatically by 2-bit stack pointer (SP).

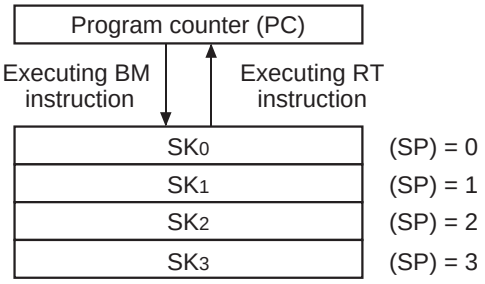
Figure 5 shows the stack registers (SKs) structure.

Figure 6 shows the example of operation at subroutine call.

(7) Skip flag

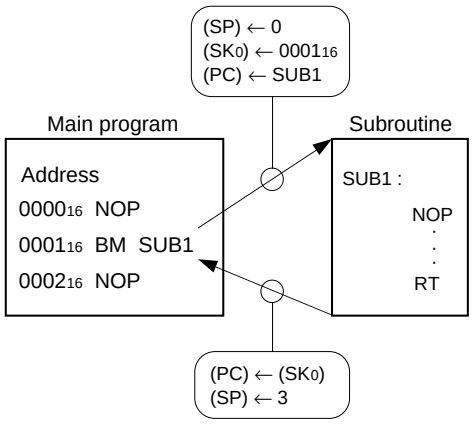
Skip flag controls skip decision for the conditional skip instructions and continuous described skip instructions.

Note : The 4283 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes "1" if the TABP p, RT, or RTS instruction is skipped.



Stack pointer (SP) points "3" at reset or returning from RAM back-up mode. It points "0" by executing the first BM instruction, and the contents of program counter is stored in SK_0.
When the BM instruction is executed after four stack registers are used ((SP) = 3), (SP) = 0 and the contents of SK_0 is destroyed.

Fig. 5 Stack registers (SKs) structure



Note: Returning to the BM instruction execution address with the RT instruction, and the BM instruction is equivalent to the NOP instruction.

Fig. 6 Example of operation at subroutine call

(8) Program counter (PC)

Program counter (PC) is used to specify a ROM address (page and address). It determines a sequence in which instructions stored in ROM are read. It is a binary counter that increments the number of instruction bytes each time an instruction is executed. However, the value changes to a specified address when branch instructions, subroutine call instructions, return instructions, or the table reference instruction (TABP p) is executed.

Program counter consists of PC_H (most significant bit to bit 7) which specifies to a ROM page and PC_L (bits 6 to 0) which specifies an address within a page. After it reaches the last address (address 127) of a page, it specifies address 0 of the next page (Figure 7).

Make sure that the PC_H does not exceed after the last page of the built-in ROM.

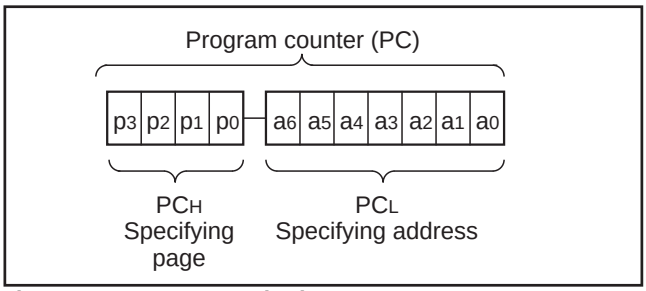


Fig. 7 Program counter (PC) structure

(9) Data pointer (DP)

Data pointer (DP) is used to specify a RAM address and consists of registers X and Y. Register X specifies a file and register Y specifies a RAM digit (Figure 8).

Register Y is also used to specify the port D bit position.

When using port D, set the port D bit position to register Y certainly and execute the SD, RD, or SZD instruction (Figure 9).

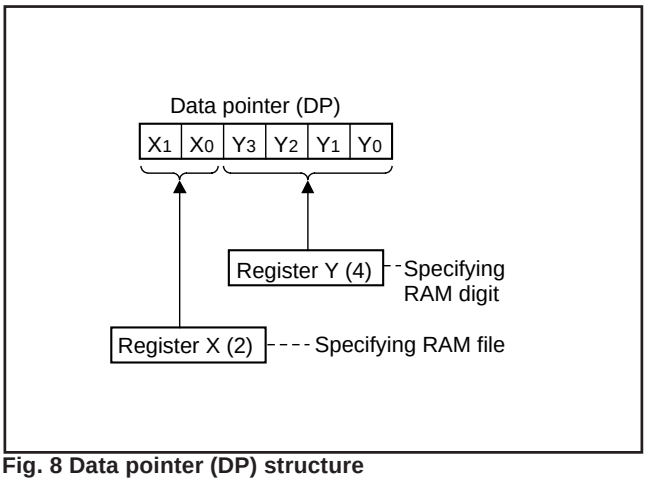


Fig. 8 Data pointer (DP) structure

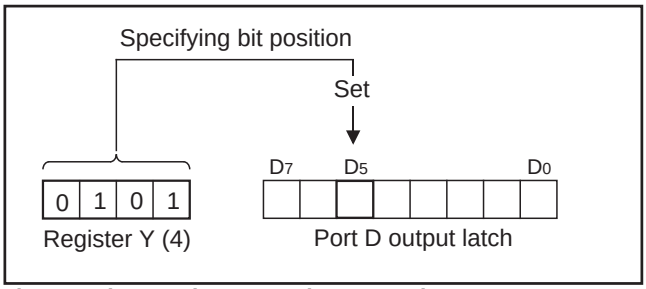


Fig. 9 SD instruction execution example

PROGRAM MEMORY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 9 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127).

Table 1 ROM size and pages

Part number	ROM size (× 9 bits)	Pages
M34283G2	2048 words	16 (0 to 15)

Page 2 (addresses 0100_{16} to $017F_{16}$) is the special page for subroutine calls. Subroutines written in this page can be called from any page with the 1-word instruction (BM). Subroutines extending from page 2 to another page can also be called with the BM instruction when it starts on page 2.
ROM pattern of all addresses can be used as data areas with the TABP p instruction.

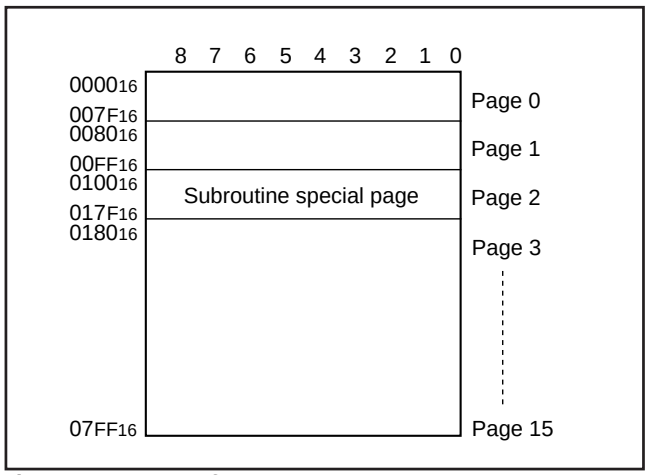


Fig. 10 ROM map of M34283G2

DATA MEMORY (RAM)

1 word of RAM is composed of 4 bits, but 1-bit manipulation (with the SB j, RB j, and SZB j instructions) is enabled for the entire memory area. A RAM address is specified by a data pointer. The data pointer consists of registers X and Y. Set a value to the data pointer certainly when executing an instruction to access RAM.
Table 2 shows the RAM size. Figure 11 shows the RAM map.

Table 2 RAM size

Part number	RAM size
M34283G2	64 words × 4 bits (256 bits)

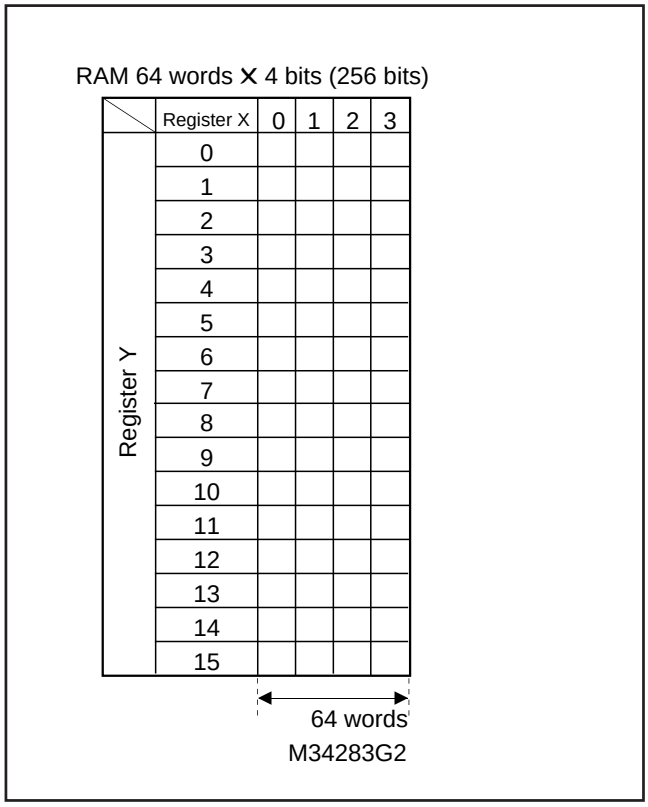


Fig. 11 RAM map

TIMERS

The 4283 Group has the programmable timer.

- Programmable timer
  The programmable timer has a reload register and enables the frequency dividing ratio to be set. It is decremented from a setting value n. When it underflows (count to n + 1), a timer 1 underflow flag is set to “1,” new data is loaded from the reload register, and count continues (auto-reload function).

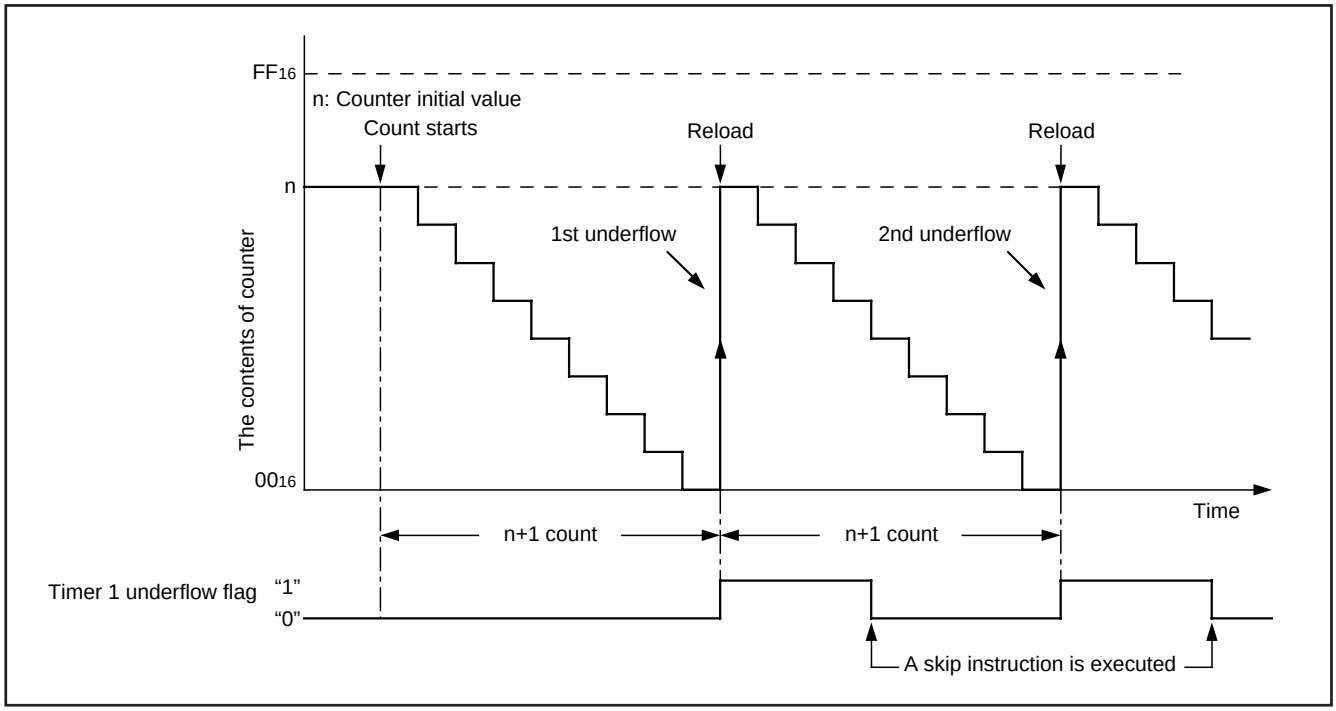


Fig. 12 Auto-reload function

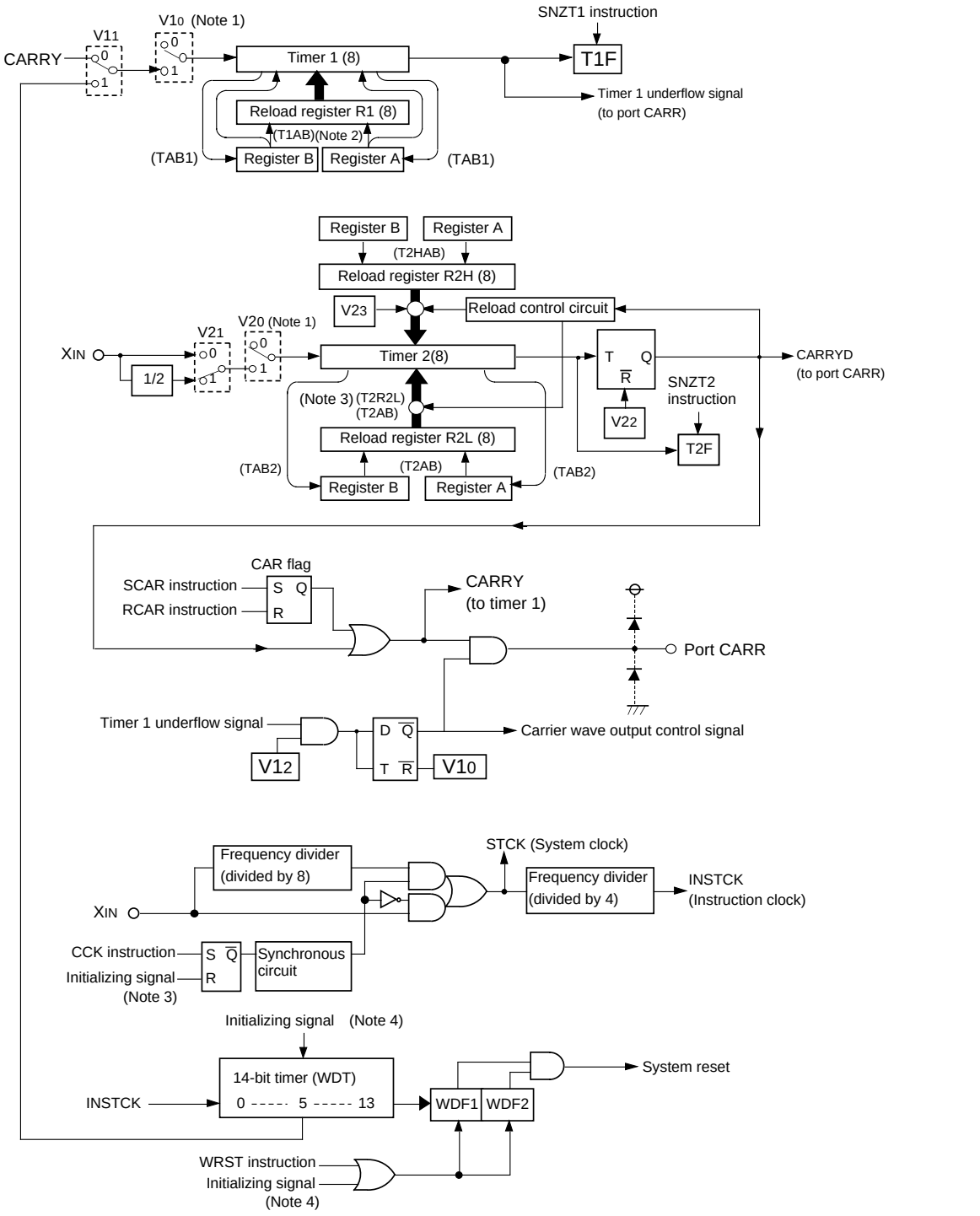
The 4283 Group timer consists of the following circuit.

- Timer 1 : 8-bit programmable timer
- Timer 2 : 8-bit programmable timer

These timers can be controlled with the timer control registers V1 and V2.
Each timer function is described below.

Table 3 Function related timer

Circuit	Structure	Count source	Frequency dividing ratio	Use of output signal	Control register
Timer 1	8-bit programmable binary down counter	• Carrier wave output (CARRY) • Bit 5 of watchdog timer	1 to 256	• Carrier wave output control	V1
Timer 2	8-bit programmable binary down counter	• $f(X_{IN})$ • $f(X_{IN})/2$	1 to 256	• Carrier wave output	V2
14-bit timer	14-bit fixed frequency	• Instruction clock	16384	• Watchdog timer • Timer 1 count source	



Notes 1: Counting is stopped by clearing to "0."
2: When the T1AB instruction is executed after V10 is set to "1," writing is performed only to reload register R1.
3: The data of reload register R2L set with the T2AB instruction can be also written to timer 2 with the T2R2L instruction.
4: The initializing signal is output at reset or RAM back-up mode.

Fig. 13 Timers structure

Table 4 Control registers related to timer

Timer control register V1		at reset : 000_2	at RAM back-up : 000_2	W
$V1_2$	Carrier wave output auto-control bit	0	Auto-control output by timer 1 is invalid	
		1	Auto-control output by timer 1 is valid	
$V1_1$	Timer 1 count source selection bit	0	Carrier wave output (CARRY)	
		1	Bit 5 of watchdog timer (WDT)	
$V1_0$	Timer 1 control bit	0	Stop (Timer 1 state retained)	
		1	Operating	

Timer control register V2		at reset : 0000_2	at RAM back-up : 0000_2	W
$V2_3$	Carrier wave "H" interval expansion bit	0	To expand "H" interval is invalid	
		1	To expand "H" interval is valid (when $V2_2$=1 selected)	
$V2_2$	Carrier wave generation function control bit	0	Carrier wave generation function invalid	
		1	Carrier wave generation function valid	
$V2_1$	Timer 2 count source selection bit	0	$f(X_{IN})$	
		1	$f(X_{IN})/2$	
$V2_0$	Timer 2 control bit	0	Stop (Timer 2 state retained)	
		1	Operating	

Note: "W" represents write enabled.

(1) Control registers related to timer

- Timer control register V1
  Register V1 controls the timer 1 count source and auto-control function of carrier wave output from port CARR by timer 1. Set the contents of this register through register A with the TV1A instruction.
- Timer control register V2
  Register V2 controls the timer 2 count source and the carrier wave generation function by timer. Set the contents of this register through register A with the TV2A instruction.

(2) Precautions

Note the following for the use of timers.

- Count source
  Stop timer 1 or timer 2 counting to change its count source.
- Reading the count value
  Stop timer 1 or 2 counting and then execute the data read instruction (TAB1, TAB2) to read its data.
- Watchdog timer
  Be sure that the timing to execute the WRST instruction in order to operate WDT efficiently.
- Writing to reload register R1
  When writing data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.
- Timer 1 count operation
  When the bit 5 of the watchdog timer (WDT) is selected as the timer 1 count source, the error of maximum ± 256 µs (at the minimum instruction execution time : 8 µs) is generated from timer 1 start until timer 1 underflow. When programming, be careful about this error.
- Stop of timer 2
  Avoid a timing when timer 2 underflows to stop timer 2.
- Writing to reload register R2H
  When writing data to reload register R2H while timer 2 is operating, avoid a timing when timer underflows.
- Timer 2 carrier wave output function
  When to expand "H" interval of carrier wave is valid, set "1" or more to reload register R2H.
- Timer 1 and timer 2 carrier wave output function
  Count starts from the rising edge ② in Fig. 14 after the first falling edge of the count source, after timer 1 and timer 2 operations start ① in Fig. 14.
  Time to first underflow ③ in Fig. 14 is different from time among next underflow ④ in Fig. 14 by the timing to start the timer and count source operations after count starts.

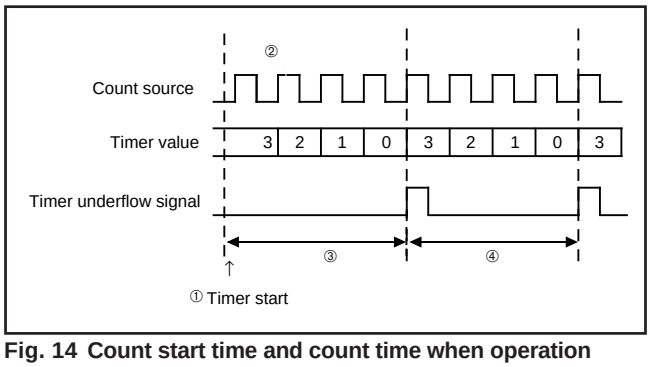


Fig. 14 Count start time and count time when operation starts (T1, T2)

(3) Timer 1

Timer 1 is an 8-bit binary down counter with the timer 1 reload register (R1).

When timer is stopped, data can be set simultaneously in timer 1 and the reload register (R1) with the T1AB instruction.

When timer is operating, data can be set to only reload register R1 with the T1AB instruction.

When setting the next count data to reload register R1 at operating, set data before timer 1 underflows.

Timer 1 starts counting after the following process;

① set data in timer 1,

② select the count source with the bit 1 of register V1, and

③ set the bit 0 of register V1 to "1."

Once count is started, when timer 1 underflows (the next count pulse is input after the contents of timer 1 becomes "0"), the timer 1 underflow flag (T1F) is set to "1," new data is loaded from reload register R1, and count continues (auto-reload function).

When a value set in reload register R1 is n, timer 1 divides the count source signal by n + 1 (n = 0 to 255).

When the bit 2 of register V1 is set to "1," the carrier wave output enable/disable interval of port CARR is alternately generated each timer 1 underflows (Figure 15).

Data can be read from timer 1 to registers A and B. When reading the data, stop the counter and then execute the TAB1 instruction.

(4) Timer 2

Timer 2 is an 8-bit binary down counter with the timer 2 reload registers (R2H and R2L).

Data can be set simultaneously in timer 2 and the reload register (R2L) with the T2AB instruction.

The contents of reload register (R2L) set with the T2AB instruction can be set again to timer 2 with the T2R2L instruction. Data can be set to reload register (R2H) with the T2HAB instruction.

Timer 2 starts counting after the following process;

① set data in timer 2,

② select the count source with the bit 1 of register V2, and

③ select the valid/invalid of the carrier wave generation function by bit 2 of register V1 (when this function is valid, select the valid/invalid of the carrier wave "H" interval expansion by bit 3), and

④ set the bit 0 of register V1 to "1."

When the carrier wave generation function is invalid ($V2_2$="0"), the following operation is performed;

Once count is started, when timer 2 underflows (the next count pulse is input after the contents of timer 2 becomes "0"), the timer 2 underflow flag (T2F) is set to "1," new data is loaded from reload register R2L, and count continues (auto-reload function).

When a value set in reload register R2L is n, timer 2 divides the count source signal by n + 1 (n = 0 to 255).

When the carrier wave generation function is valid ($V2_2$="1"), the carrier wave which has the "L" interval set to the reload register R2L and "H" interval set to the reload register R2H can be output (Figure 16).

After the count of the "L" interval of carrier wave is started, timer 2 underflows and the timer 2 underflow flag (T2F) is set to "1". Then, the "H" interval data of carrier wave is reloaded from the reload register R2H, and count continues.

When timer underflows again after auto-reload, the T2F flag is set to "1". And then, the "L" interval data of carrier wave is reloaded from the reload register R2L, and count continues. After that, each timer underflows, data is reloaded from reload register R2H and R2L alternately.

When a value set in reload register R2H is n, "H" interval of carrier wave is as follows;

① When to expand "H" interval is invalid ($V2_3$ = "0"),
Count source × (n+1), n = 0 to 255

② When to expand "H" interval is valid ($V2_3$ = "1"),
Count source × (n+1.5), n = 1 to 255

When a value set in reload register R2L is m, "L" interval of carrier wave is as follows;

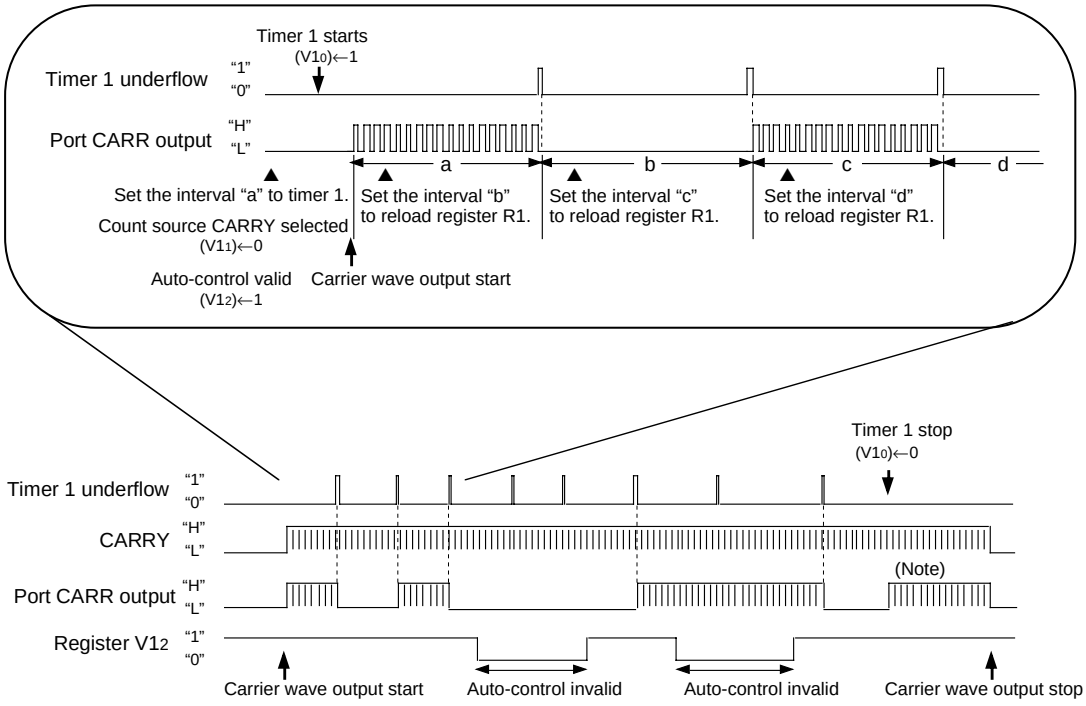
Count source × (m+1), m = 0 to 255

Data can be read from timer 2 to registers A and B. When reading the data, stop the counter and then execute the TAB2 instruction.

(5) Timer underflow flags (T1F, T2F)

Timer 1 underflow flag or timer 2 underflow flag is set to "1" when the timer 1 or timer 2 underflows. The state of flags T1F and T2F can be examined with the skip instruction (SNZT1, SNZT2).

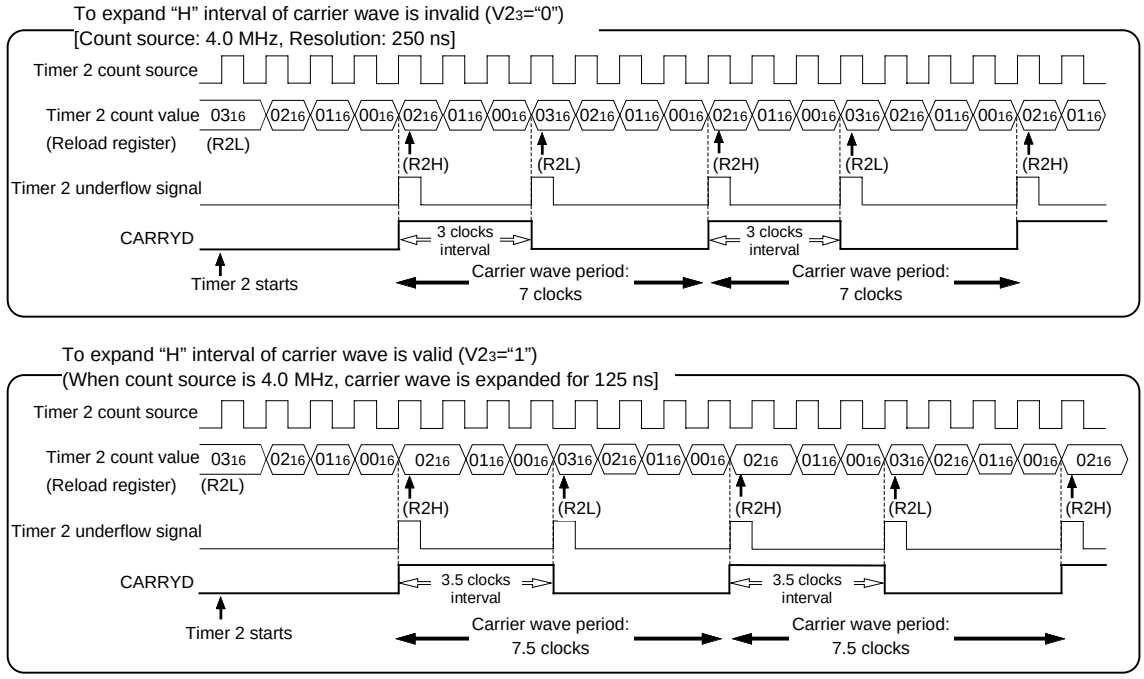
Flags T1F and T2F are cleared to "0" when the next instruction is skipped with a skip instruction.



Note: When timer 1 is stopped, the port CARR output auto-control is terminated regardless of bit 2 (V12) of register V1.

Fig. 15 Port CARR output control by timer 1

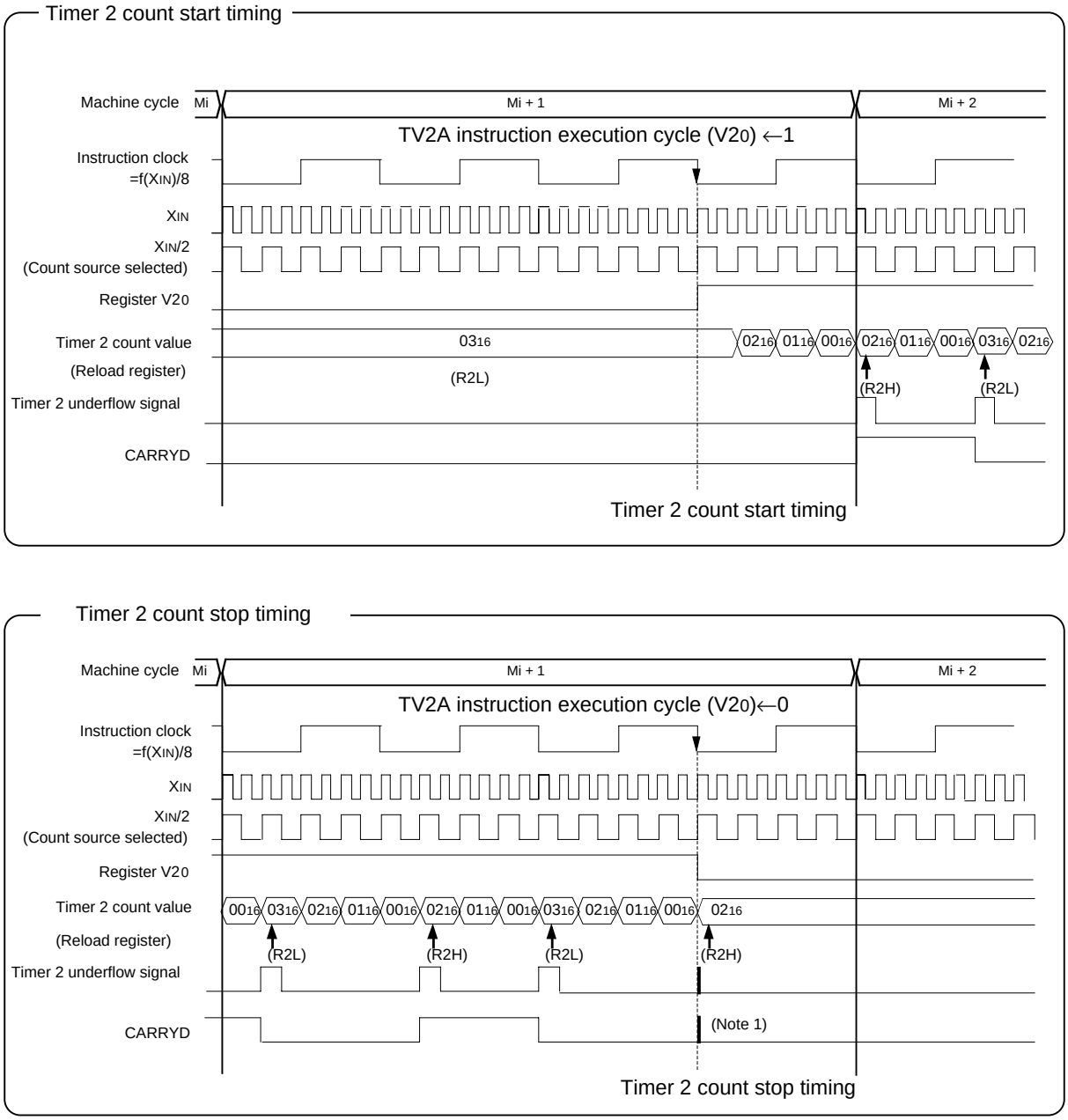
● In this case, the following is set;
- Timer 2 carrier wave generation function is valid (V22="1"),
- "L" interval (03_{16}) of carrier wave is set to reload register R2L
- "H" interval (02_{16}) of carrier wave is set to reload register R2H



Note: When to expand "H" interval of the carrier wave is valid, set "01_{16}" or more to reload register R2H.

Fig. 16 Carrier wave generation example by timer 2

● In this case, the following is set;
- To expand “H” interval of carrier wave is invalid (V23 = “0”),
- Timer 2 carrier wave generation function is valid (V22=“1”),
- Count source XIN/2 selected (V21=“1”),
- “L” interval (0316) of carrier wave is set to reload register R2L
- “H” interval (0216) of carrier wave is set to reload register R2H



Notes 1: When the carrier wave generation function is vaild (V22=“1”), avoid a timing when timer 2 underflows to stop timer 2. When the timer 2 count stop occurs at the same timing with the timer 2 underflows, hazard may occur in the carrier wave output waveform.

2: When the timer 2 is stopped during “H” output of carrier wave while the carrier wave generation function is valid, it is stopped after the “H” interval set by reload register R2H is output.

Fig. 17 Timer 2 count start/stop timing

WATCHDOG TIMER

Watchdog timer provides a method to reset and restart the system when a program runs wild. Watchdog timer consists of 14-bit timer (WDT) and watchdog timer flags (WDF1, WDF2).
Watchdog timer downcounts the instruction clock (INSTCK) as the count source immediately after system is released from reset. When the timer WDT count value becomes 0000_{16} and underflow occurs, the WDF1 flag is set to "1." Then, when the WRST instruction is not executed before the timer WDT counts 16383, WDF2 flag is set to "1" and internal reset signal is generated and system reset is performed.
Execute the WRST instruction at period of 16383 machine cycle or less to keep the microcomputer operation normal.
Timer WDT is also used for generation of oscillation stabilization time. When system is returned from reset and from RAM back-up mode by key-input, software starts after the stabilization oscillation time until timer WDT downcounts to $3E00_{16}$ elapses.

- Watchdog timer
  Be sure that the timing to execute the WRST instruction in order to operate WDT efficiently.

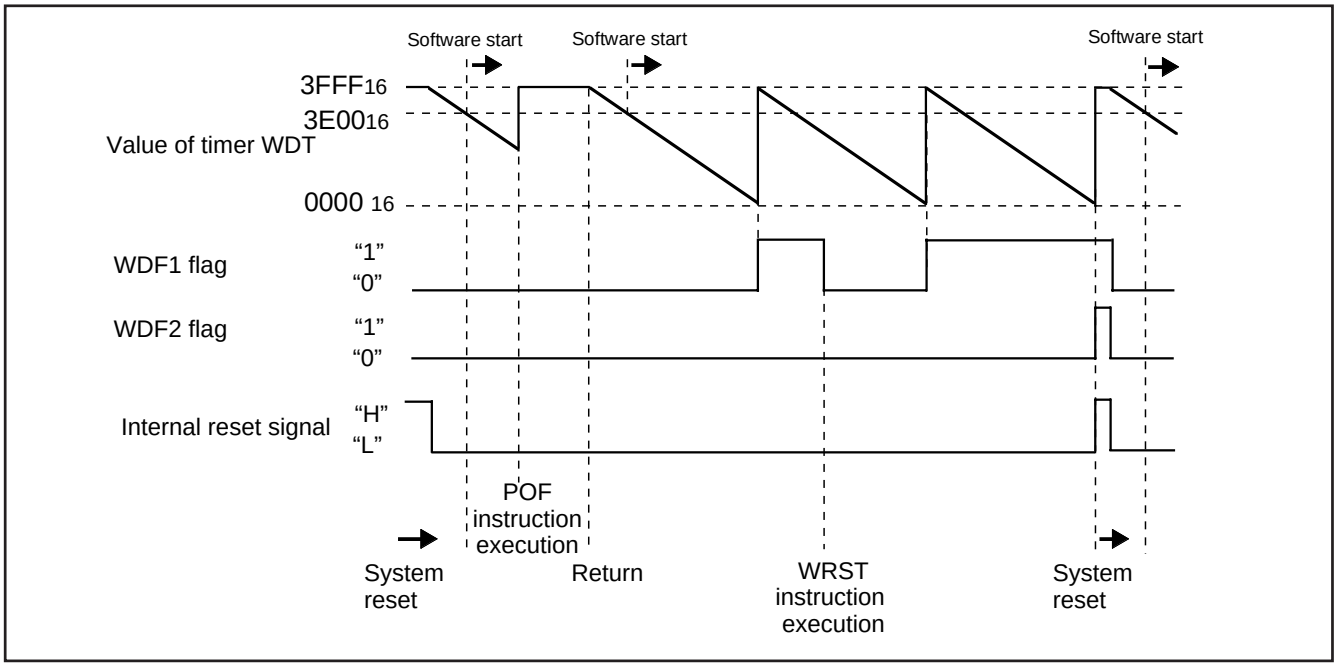


Fig. 18 Watchdog timer function

LOGIC OPERATION FUNCTION

The 4283 Group has the 4-bit logic operation function. The logic operation between the contents of register A and the low-order 4 bits of register E is performed and its result is stored in register A.

Each logic operation can be selected by setting logic operation selection register LO.
Set the contents of this register through register A with the TLOA instruction. The logic operation selected by register LO is executed with the LGOP instruction.
Table 5 shows the logic operation selection register LO.

Table 5 Logic operation selection register LO

Logic operation selection register LO		at reset : 00_2		at RAM back-up : 00_2	W
LO_1	Logic operation selection bits	LO_1	LO_0	Logic operation function	
		0	0	Exclusive logic OR operation (XOR)	
LO_0		0	1	OR operation (OR)	
		1	0	AND operation (AND)	
		1	1	Not available	

Note: "W" represents write enabled.

RESET FUNCTION

The 4283 Group has the power-on reset circuit, though it does not have $\overline{\text{RESET}}$ pin. System reset is performed automatically at power-on, and software starts program from address 0 in page 0.

In order to make the built-in power-on reset circuit operate efficiently, set the voltage rising time until V_{DD}= 0 to 2.2 V is obtained at power-on 1ms or less.

Note on Power-on reset

Under the following condition, the system reset occurs by the built-in the power-on reset circuit of this product;

- when the supply voltage (V_{DD}) rises from 0 V to 2.2 V, within 1 ms.

  Also, note that system reset does not occur under the following conditions;
- when the supply voltage (V_{DD}) rises from the voltage higher than 0V, or
- when it takes more than 1 ms for the supply voltage (V_{DD}) to rise from 0 V to 2.2 V.

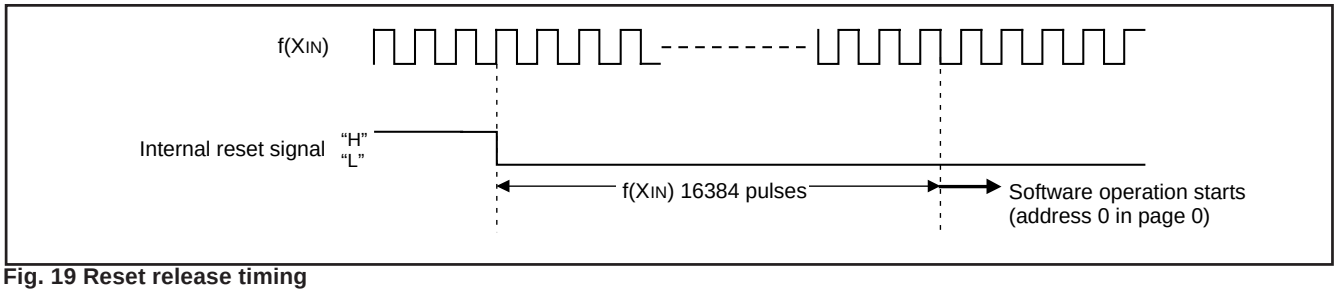


Fig. 19 Reset release timing

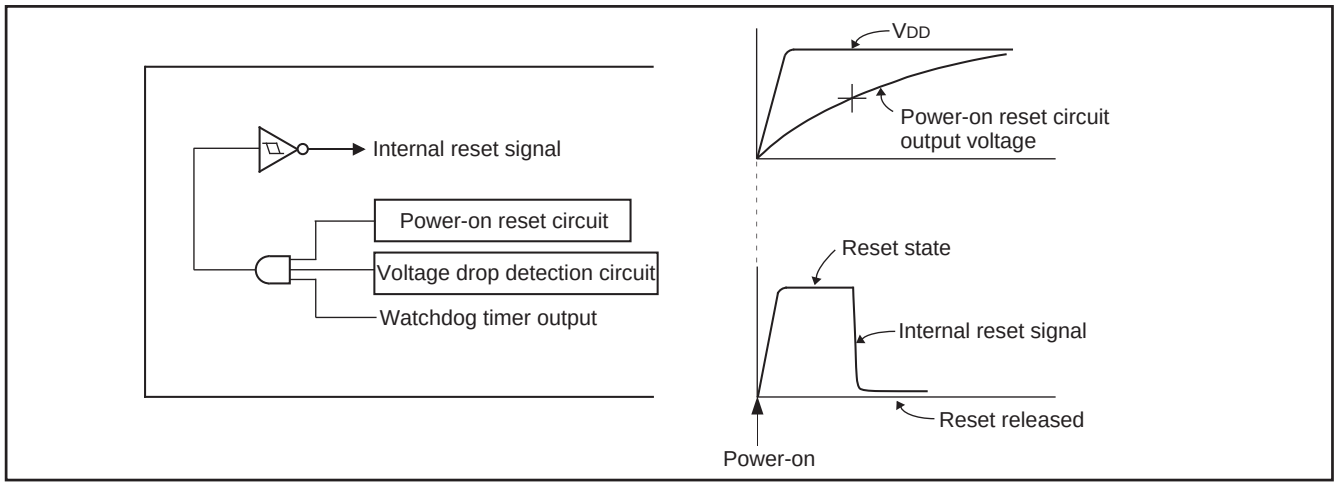


Fig. 20 Structure of reset pin and its peripherals, and power-on reset operation

(1) Internal state at reset

Table 6 shows port state at reset, and Figure 21 shows internal state at reset (they are retained after system is released from reset).

The contents of timers, registers, flags and RAM except shown in Figure 21 are undefined, so set the initial value to them.

Table 6 Port state at reset

Name	State at reset
D_0–D_3	High impedance state
D_4–D_7	High impedance state (Pull-down transistor OFF)
G_0–G_3	High impedance state (Pull-down transistor OFF)
E_0, E_1	High impedance state (Pull-down transistor OFF)
CARR	"L" output

Note: The contents of all output latch is initialized to "0."

- Program counter (PC) 0 0 0 0 0 0 0 0 0 0 0 0
  Address 0 in page 0 is set to program counter.
- Power down flag (P) 0
- Timer 1 underflow flag (T1F) 0
- Timer 2 underflow flag (T2F) 0
- Timer control register V1 ..... 0 0 0
- Timer control register V2 ..... 0 0 0 0
- Port CARR output flag (CAR) 0
- Pull-down control register PU0 ..... 0 0 0 0
- Pull-down control register PU1 ..... 0 0 0 0
- Logic operation selection register LO 0 0
- Most significant ROM code reference enable flag (URS) 0
- Carry flag (CY) 0
- Register A 1 1 1 1
- Register B 1 1 1 1
- Register X ..... X X
- Register Y ..... X X X X
- Stack pointer (SP) 1 1

"X" represents undefined.

Fig. 21 Internal state at reset

VOLTAGE DROP DETECTION CIRCUIT

The built-in voltage drop detection circuit is designed to detect a drop in voltage at operating and to reset the microcomputer if the supply voltage drops below the specified value (Typ. 1.50 V) or less.

The voltage drop detection circuit is stopped and power dissipation is reduced in the RAM back-up mode with the initialized CPU stopped.

Note on voltage drop detection circuit

The voltage drop detection circuit detection voltage of this product is set up lower than the minimum value of the supply voltage of the recommended operating conditions.

A battery exchange of an application product is explained as an example.

The supply voltage falls below to the recommended operating voltage while CPU keeps active. Then, an unexpected oscillation-stop, which does not happen by POF instruction occurs before the supply voltage falls below to the detection voltage. In this time, even if the supply voltage re-goes up to the recommended operating voltage, since reset does not occur, MCU may not operate correctly.

Please confirm the oscillator you use and the frequency of system clock, and test the operation of your system sufficiently.

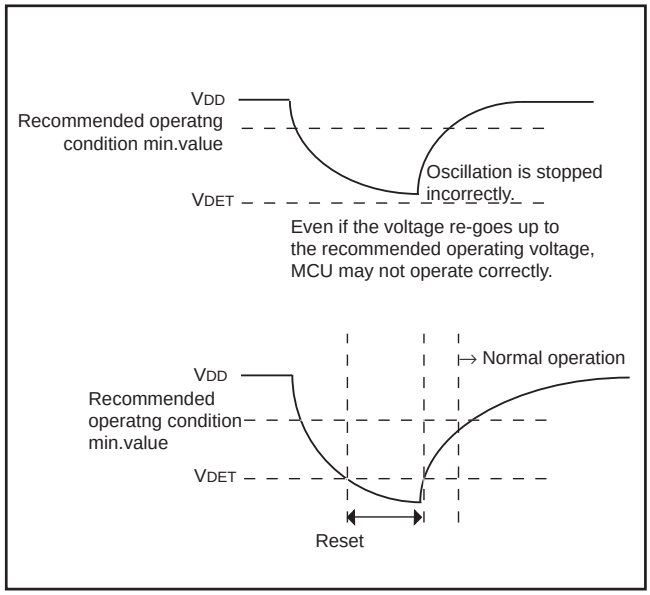


Fig. 23 VDD and VDET

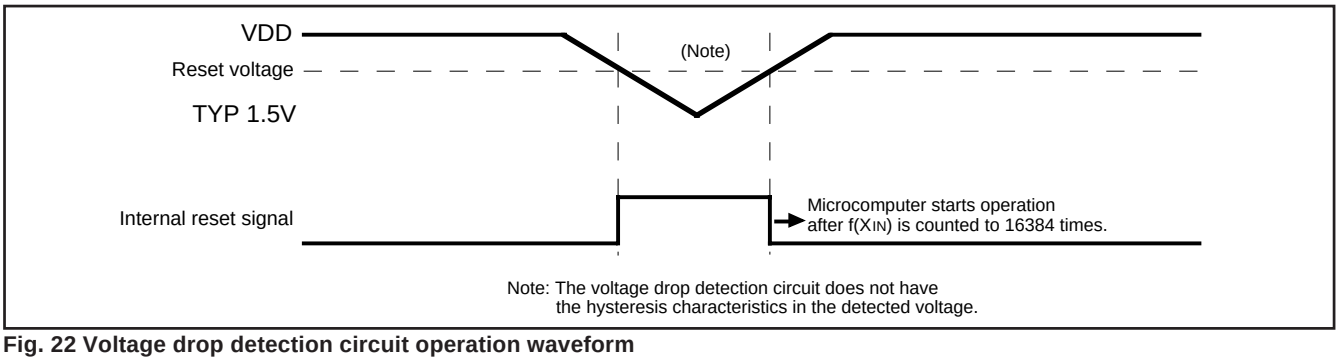


Fig. 22 Voltage drop detection circuit operation waveform

RAM BACK-UP MODE

The 4283 Group has the RAM back-up mode.
When the POF instruction is executed, system enters the RAM back-up state.
As oscillation stops retaining RAM, the functions and states of reset circuit at RAM back-up mode, power dissipation can be reduced without losing the contents of RAM. Table 7 shows the function and states retained at RAM back-up. Figure 24 shows the state transition.

(1) Warm start condition

When the external wakeup signal is input after the system enters the RAM back-up state by executing the POF instruction, the CPU starts executing the software from address 0 in page 0. In this case, the P flag is "1."

(2) Cold start condition

The CPU starts executing the software from address 0 in page 0 when any of the following conditions is satisfied .

- reset by power-on reset circuit is performed
- reset by watchdog timer is performed
- reset by voltage drop detection circuit is performed

In this case, the P flag is "0."

(3) Identification of the start condition

Warm start (return from the RAM back-up state) or cold start (return from the normal reset state) can be identified by examining the state of the power down flag (P) with the SNZP instruction.

Table 7 Functions and states retained at RAM back-up

Function	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2)	×
Contents of RAM	O
Port CARR	×
Ports D_0–D_7	O
Ports E_0, E_1	O
Port G	O
Timer control registers V1, V2	×
Pull-down control registers PU0, PU1	O
Logic operation selection register LO	×
Timer 1 function, Timer 2 function	×
Timer underflow flags (T1F, T2F)	×
Watchdog timer (WDT)	×
Watchdog timer flags (WDF1, WDF2)	×
Most significant ROM code reference enable flag (URS)	×

Notes 1: "O" represents that the function can be retained, and "×" represents that the function is initialized.
Registers and flags other than the above are undefined at RAM back-up, and set an initial value after returning.
2:The stack pointer (SP) points the level of the stack register and is initialized to "11_2" at RAM back-up.

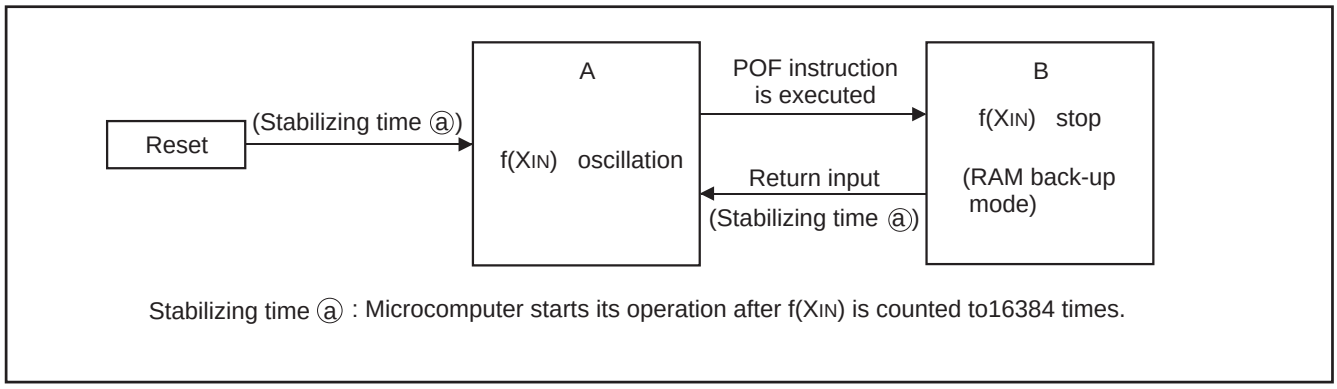


Fig. 24 State transition

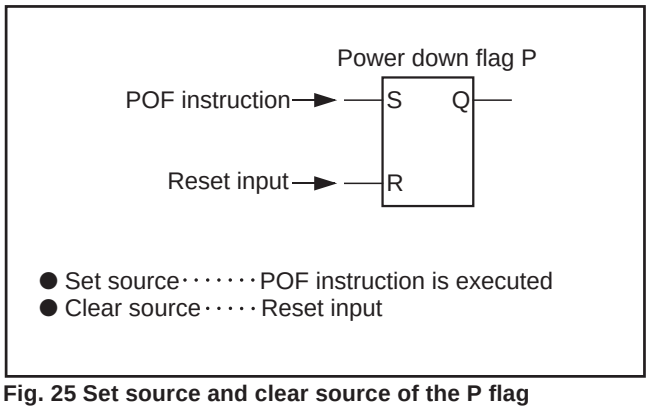


Fig. 25 Set source and clear source of the P flag

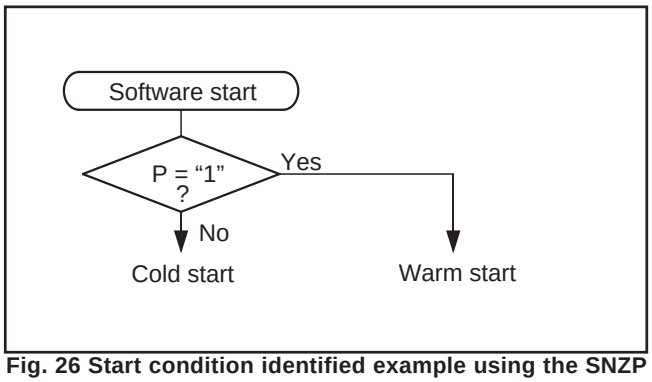


Fig. 26 Start condition identified example using the SNZP instruction

(4) Return signal

An external wakeup signal is used to return from the RAM back-up mode. Table 8 shows the return condition for each return source.

Table 8 Return source and return condition

Return source	Return condition	Remarks
Ports D_4–D_7	Return by an external "H" level input.	Only key-on wakeup function of the port whose pull-down transistor is turned ON by register PU1 is valid.
Ports E_0, E_1, G	Return by an external "H" level input.	Only key-on wakeup function of the port whose pull-down transistor is turned ON by register PU0 is valid.
Port E_2	Return by an external "H" level input.	Key-on wakeup function is always valid.

(5) Pull-down control register

Registers PU0 and PU1 are 4-bit registers and control the ON/OFF of pull-down transistor and key-on wakeup function for ports E_0, E_1, G and ports D_4–D_7.

Set the contents of register PU0 or PU1 through register A with the TPU0A or TPU1A instruction, respectively.

Table 9 Pull-down control registers

Pull-down control register PU0		at reset : 0000_2		at RAM back-up : state retained	W
$PU0_3$	Ports G_2, G_3 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU0_2$	Ports G_0, G_1 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU0_1$	Port E_1 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU0_0$	Port E_0 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		

Pull-down control register PU1		at reset : 0000_2		at RAM back-up : state retained	W
$PU1_3$	Port D_7 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU1_2$	Port D_6 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU1_1$	Port D_5 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU1_0$	Port D_4 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		

Note: "W" represents write enabled.

CLOCK CONTROL

The clock control circuit consists of the following circuits.

- System clock generating circuit
- Control circuit to stop the clock oscillation
- Control circuit to return from the RAM back-up state

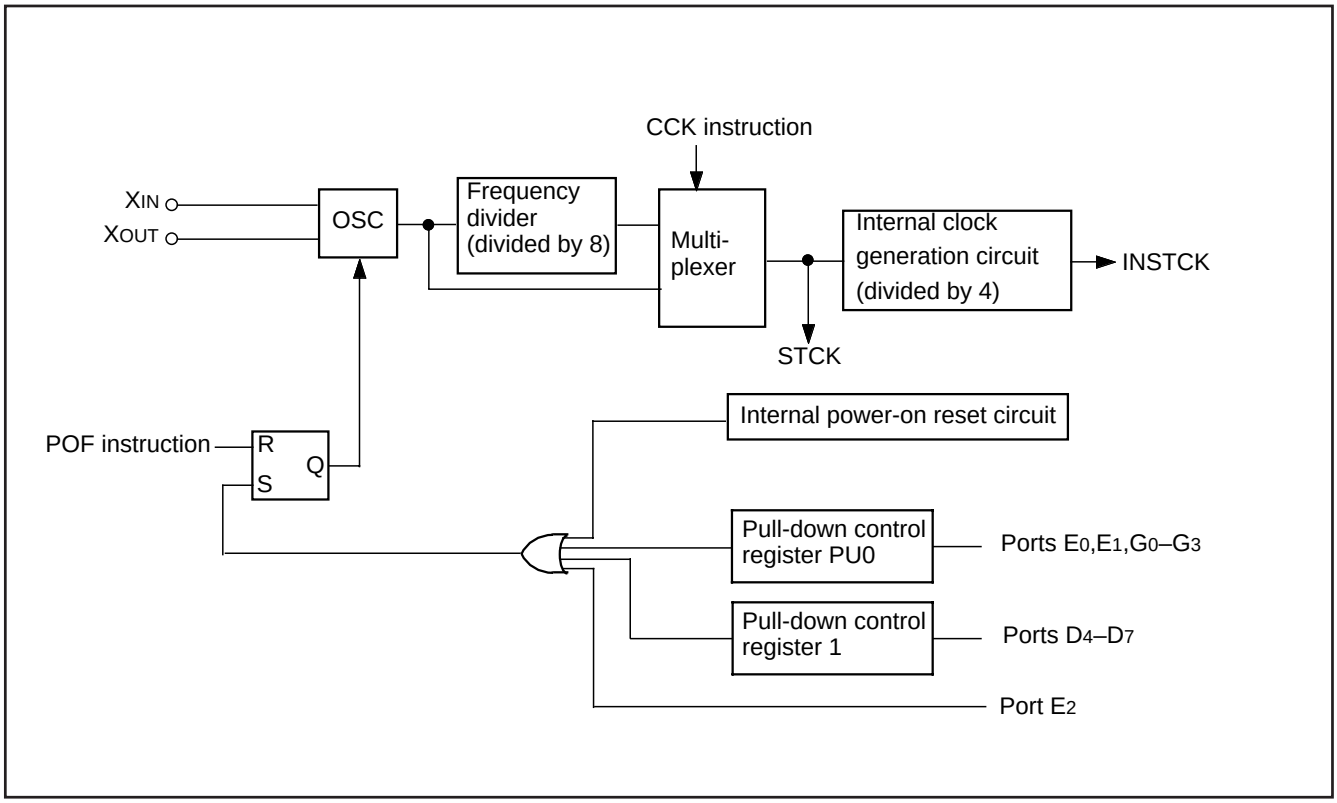


Fig. 27 Clock control circuit structure

System clock signal f(XIN) is obtained by externally connecting a ceramic resonator. Connect this external circuit to pins XIN and XOUT at the shortest distance as shown Figure 28.
A feedback resistor is built-in between XIN pin and XOUT pin.

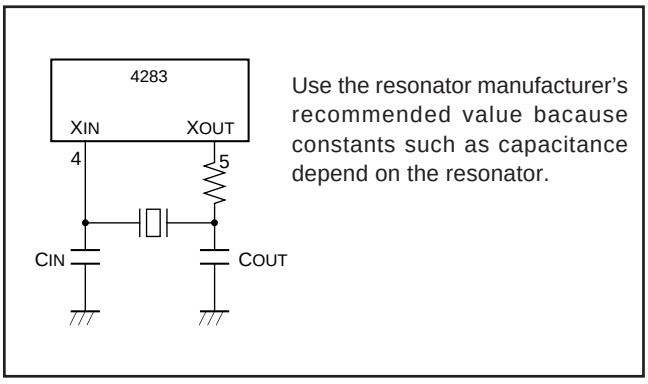


Fig. 28 Ceramic resonator external circuit

LIST OF PRECAUTIONS

① Noise and latch-up prevention

Connect a capacitor on the following condition to prevent noise and latch-up;

- connect a bypass capacitor (approx. 0.01 μF) between pins V_{DD} and V_{SS} at the shortest distance,
- equalize its wiring in width and length, and
- use the thickest wire.
- Port E_2 is also uesd as V_{PP} pin. Connect this pin to V_{SS} through the resistor about 5kΩ which is assigned to E_2/V_{PP} pin as close as possible at the shortest distance.

② Register initial values 1

The initial value of the following registers are undefined after system is released from reset. After system is released from reset, set initial values.

- Register D (3 bits)
- Register E (8 bits)

③ Register initial values 2

The initial value of the following registers are undefined at RAM backup. After system is returned from RAM back-up, set initial values.

- Register X (4 bits)
- Register Y (4 bits)
- Register D (3 bits)
- Register E (8 bits)

④ Stack registers (SK_S)

Stack registers (SK_S) are four identical registers, so that subroutines can be nested up to 4 levels. However, one of stack registers is used respectively when using an interrupt service routine and when executing a table reference instruction. Accordingly, be careful not to over the stack when performing these operations together.

⑤ Notes on unused pins

Pin	Connection	Usage condition
D_0–D_3	Open.	
	Connect to V_{DD}.	
D_4–D_7	Open (Set the output latch to “1”).	Pull-down transistor OFF.
	Open (Set the output latch to “0”).	
	Connect to V_{DD}.	Pull-down transistor OFF.
E_0, E_1	Open (Set the output latch to “1”).	Pull-down transistor OFF.
	Open (Set the output latch to “0”).	
	Connect to V_{DD}.	Pull-down transistor OFF.
E_2	Open.	
	Connect to V_{SS}.	
G_0–G_3	Open (Set the output latch to “1”).	Pull-down transistor OFF.
	Open (Set the output latch to “0”).	
	Connect to V_{DD}.	Pull-down transistor OFF.
CARR	Open.	

(Note when connecting to V_{SS} and V_{DD})

- Connect the unused pins to V_{SS} and V_{DD} at the shortest distance and use the thick wire against noise.

⑥ Timer

- Count source
  Stop timer 1 or timer 2 counting to change its count source.
- Reading the count value
  Stop timer 1 or 2 counting and then execute the data read instruction (TAB1, TAB2) to read its data.
- Watchdog timer
  Be sure that the timing to execute the WRST instruction in order to operate WDT efficiently.
- Writing to reload register R1
  When writing data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.
- Timer 1 count operation
  When the bit 5 of the watchdog timer (WDT) is selected as the timer 1 count source, the error of maximum ± 256 µs (at the minimum instruction execution time : 8 µs) is generated from timer 1 start until timer 1 underflow. When programming, be careful about this error.
- Stop of timer 2
  Avoid a timing when timer 2 underflows to stop timer 2.
- Writing to reload register R2H
  When writing data to reload register R2H while timer 2 is operating, avoid a timing when timer underflows.
- Timer 2 carrier wave output function
  When to expand “H” interval of carrier wave is valid, set “1” or more to reload register R2H.
- Timer 1 and timer 2 carrier wave output function
  Count starts from the rising edge ② in Fig. 29 after the first falling edge of the count source, after timer 1 and timer 2 operations start ① in Fig. 29.
  Time to first underflow ③ in Fig. 29 is different from time among next underflow ④ in Fig. 29 by the timing to start the timer and count source operations after count starts.

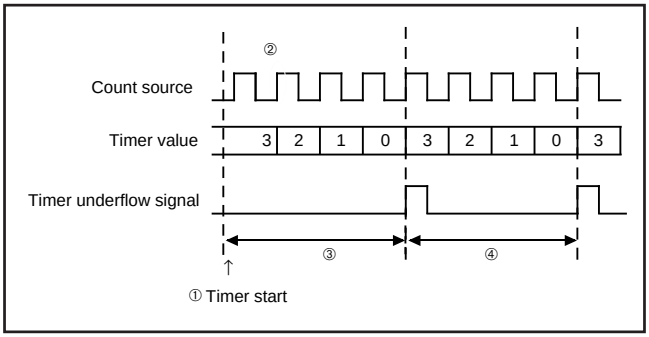


Fig. 29 Count start time and count time when operation starts (T1, T2)

⑦ Program counter

Make sure that the program counter does not specify after the last page of the built-in ROM.

⑧ Power-on reset

Under the following condition, the system reset occurs by the built-in the power-on reset circuit of this product;

- when the supply voltage (V_{DD}) rises from 0 V to 2.2 V, within 1 ms.
  Also, note that system reset does not occur under the following conditions;
- when the supply voltage (V_{DD}) rises from the voltage higher than 0V, or
- when it takes more than 1 ms for the supply voltage (V_{DD}) to rise from 0 V to 2.2 V.

⑨ Voltage drop detection circuit

The voltage drop detection circuit detection voltage of this product is set up lower than the minimum value of the supply voltage of the recommended operating conditions.

A battery exchange of an application product is explained as an example.

The supply voltage falls below to the recommended operating voltage while CPU keeps active. Then, an unexpected oscillation-stop, which does not happen by POF instruction occurs before the supply voltage falls below to the detection voltage. In this time, even if the supply voltage re-goes up to the recommended operating voltage, since reset does not occur, MCU may not operate correctly.

Please confirm the oscillator you use and the frequency of system clock, and test the operation of your system sufficiently.

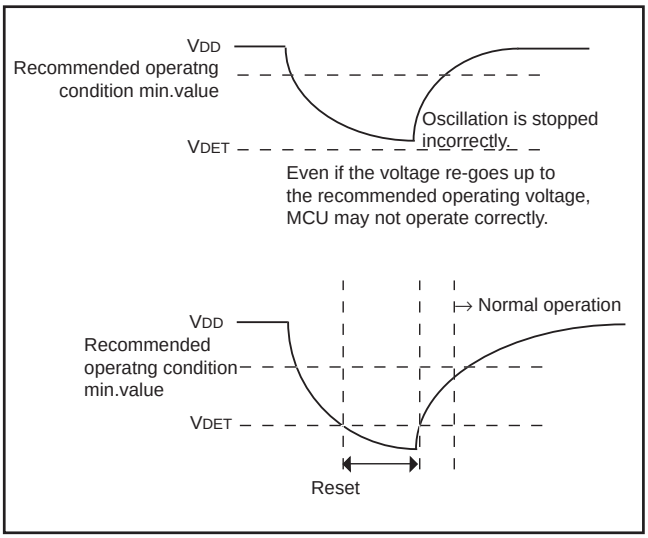


Fig. 30 V_{DD} and V_{DET}

⑩ Note on Power Source Voltage

When the power source voltage value of a microcomputer is less than the value which is indicated as the recommended operating conditions, the microcomputer does not operate normally and may perform unstable operation.

In a system where the power source voltage drops slowly when the power source voltage drops or the power supply is turned off, reset a microcomputer when the supply voltage is less than the recommended operating conditions and design a system not to cause errors to the system by this unstable operation.

⑪ Note on product shipped in blank

As for the product shipped in blank, Renesas does not perform the writing test to user ROM area after the assembly process though the QzROM writing test is performed enough before the assembly process. Therefore, a writing error of approx. 0.1 % may occur.

Moreover, please note the contact of cables and foreign bodies on a socket, etc. because a writing environment may cause some writing errors.

⑫ QzROM

(1) Be careful not to apply overvoltage to MCU. The contents of QzROM may be overwritten because of overvoltage. Take care especially at turning on the power.

(2) As for the product shipped in blank, Renesas does not perform the writing test to user ROM area after the assembly process though the QzROM writing test is performed enough before the assembly process. Therefore, a writing error of approx.0.1 % may occur. Moreover, please note the contact of cables and foreign bodies on a socket, etc. because a writing environment may cause some writing errors.

⑬ Notes On ROM Code Protect

(QzROM product shipped after writing)

As for the QzROM product shipped after writing, the ROM code protect is specified according to the ROM option setup data in the mask file which is submitted at ordering.

The ROM option setup data in the mask file is “00_{16}” for protect enabled or “FF_{16}” for protect disabled.

Note that the mask file which has nothing at the ROM option data or has the data other than “00_{16}” and “FF_{16}” can not be accepted.

INSTRUCTIONS

The 4283 Group has the 68 instructions. Each instruction is described as follows;
(1) List of instruction function
(2) Machine instructions (index by alphabet)
(3) Machine instructions (index by function)
(4) Instruction code table

SYMBOL

The symbols shown below are used in the following list of instruction function and the machine instructions.

Symbol	Contents	Symbol	Contents
A	Register A (4 bits)	D	Port D (8 bits)
B	Register B (4 bits)	E	Port E (3 bits)
DR	Register D (3 bits)	G	Port G (4 bits)
ER	Register E (8 bits)	CARR	Port CARR (1 bit)
V1	Timer control register V1 (3 bits)	CAR	CAR flag (1 bit)
V2	Timer control register V2 (4 bits)		
PU0	Pull-down control register PU0 (4 bits)	x	Hexadecimal variable
PU1	Pull-down control register PU1 (4 bits)	y	Hexadecimal variable
LO	Logic operation selection register LO (2 bits)	p	Hexadecimal variable
		n	Hexadecimal constant which represents the
X	Register X (2 bits)		immediate value
Y	Register Y (4 bits)	j	Hexadecimal constant which represents the
DP	Data pointer (6 bits)		immediate value
	(It consists of registers X and Y)	$A_3A_2A_1A_0$	Binary notation of hexadecimal variable A
PC	Program counter (11 bits)		(same for others)
PC_H	High-order 4 bits of program counter		
PC_L	Low-order 7 bits of program counter	←	Direction of data movement
SK	Stack register (11 bits × 4)	↔	Data exchange between a register and memory
SP	Stack pointer (2 bits)	?	Decision of state shown before "?"
CY	Carry flag	( )	Contents of registers and memories
R1	Timer 1 reload register	—	Negate, Flag unchanged after executing
T1	Timer 1		instruction
T1F	Timer 1 underflow flag	M(DP)	RAM address pointed by the data pointer
R2H	Timer 2 reload register	a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
R2L	Timer 2 reload register	p, a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
T2	Timer 2		in page p_3 p_2 p_1 p_0
T2F	Timer 2 underflow flag	C	Hex. number C + Hex. number x (also same for
WDT	Watchdog timer	+	others)
WDF1	Watchdog timer flag 1	x	
WDF2	Watchdog timer flag 2		
URS	Most significant ROM code reference enable flag		
P	Power down flag		
STCK	System clock		
INSTCK	Instruction clock		

Note : The 4283 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes "1" if the TABP p, RT, or RTS instruction is skipped.

LIST OF INSTRUCTION FUNCTION

Grouping	Mnemonic	Function	Page
Register to register transfer	TAB	(A) ← (B)	40
	TBA	(B) ← (A)	42
	TAY	(A) ← (Y)	42
	TYA	(Y) ← (A)	44
	TEAB	(ER_7–ER_4) ← (B) (ER_3–ER_0) ← (A)	43
	TABE	(B) ← (ER_7–ER_4) (A) ← (ER_3–ER_0)	41
	TDA	(DR_2–DR_0) ← (A_2–A_0)	42
RAM addresses	LXY x, y	(X) ← x, x = 0 to 3 (Y) ← y, y = 0 to 15	33
	INY	(Y) ← (Y) + 1	33
	DEY	(Y) ← (Y) – 1	32
RAM to register transfer	TAM j	(A) ← (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3	42
	XAM j	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3	45
	XAMD j	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) – 1	45
	XAMI j	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) + 1	45

Grouping	Mnemonic	Function	Page
Arithmetic operation	LA n	(A) ← n n = 0 to 15	33
	TABP p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p p=0 to 15 (PC_L) ← (DR_2–DR_0, A_3–A_0) When URS=0 (B) ← (ROM(PC))$_{7 \text{ to } 4}$ (A) ← (ROM(PC))$_{3 \text{ to } 0}$ When URS=1 (CY) ← (ROM(PC))$_8$ (B) ← (ROM(PC))$_{7 \text{ to } 4}$ (A) ← (ROM(PC))$_{3 \text{ to } 0}$ (PC) ← (SK(SP)) (SP) ← (SP) – 1	41
	AM	(A) ← (A) + (M(DP))	29
	AMC	(A) ← (A) + (M(DP)) + (CY) (CY) ← Carry	29
	A n	(A) ← (A) + n n = 0 to 15	29
	SC	(CY) ← 1	37
	RC	(CY) ← 0	35
	SZC	(CY) = 0 ?	39
	CMA	(A) ← ($\overline{A}$)	32
	RAR	→ CY → $A_3A_2A_1A_0$	35
	LGOP	Logic operation instruction XOR, OR, AND	33
Bit operation	SB j	(Mj(DP)) ← 1 j = 0 to 3	36
	RB j	(Mj(DP)) ← 0 j = 0 to 3	35
	SZB j	(Mj(DP)) = 0 ? j = 0 to 3	39

Grouping	Mnemonic	Function	Page
Comparison operation	SEAM	(A) = (M(DP)) ?	38
	SEA n	(A) = n ? n = 0 to 15	37
Branch operation	B a	(PC_L) ← a_6–a_0	29
	BL p, a	(PC_H) ← p (PC_L) ← a_6–a_0	30
	BA a	(PC_L) ← (a_6–a_4, A_3–A_0)	30
	BLA p, a	(PC_H) ← p (PC_L) ← (a_6–a_4, A_3–A_0)	30
Subroutine operation	BM a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← 2 (PC_L) ← a_6–a_0	30
	BML p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p p= 0 to 15 (PC_L) ← a_6–a_0	31
	BMLA p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p p= 0 to 15 (PC_L) ← (a_6–a_4, A_3–A_0)	31
Return operation	RT	(PC) ← (SK(SP)) (SP) ← (SP) – 1	36
	RTS	(PC) ← (SK(SP)) (SP) ← (SP) – 1	36

Grouping	Mnemonic	Function	Page
Timer operation	TV1A	($V1_2$–$V1_0$) ← (A_2–A_0)	44
	TAB1	(B) ← ($T1_7$–$T1_4$) (A) ← ($T1_3$–$T1_0$)	41
	T1AB	at timer 1 stop ($V1_0$=0): ($R1_7$–$R1_4$) ← (B) ($T1_7$–$T1_4$) ← (B) ($R1_3$–$R1_0$) ← (A) ($T1_3$–$T1_0$) ← (A) at timer 1 operating ($V1_0$=1): ($R1_7$–$R1_4$) ← (B) ($R1_3$–$R1_0$) ← (A)	39
	SNZT1	(T1F) = 1 ? (T1F) ← 0	38
	TV2A	($V2_3$–$V2_0$) ← (A_3–A_0)	44
	TAB2	(B) ← ($T2_7$–$T2_4$) (A) ← ($T2_3$–$T2_0$)	41
	T2AB	($R2L_7$–$R2L_4$) ← (B) ($T2_7$–$T2_4$) ← (B) ($R2L_3$–$R2L_0$) ← (A) ($T2_3$–$T2_0$) ← (A)	40
	T2HAB	($R2H_7$–$R2H_4$) ← (B) ($R2H_3$–$R2H_0$) ← (A)	40
	T2R2L	($T2_7$–$T2_4$) ← ($R2L_7$–$R2L_4$) ($T2_3$–$T2_0$) ← ($R2L_3$–$R2L_0$)	40
	SNZT2	(T2F) = 1 ? (T2F) ← 0	38

LIST OF INSTRUCTION FUNCTION (CONTINUED)

Grouping	Mnemonic	Function	Page
Input/Output operation	CLD	(D) ← 0	31
	RD	(D(Y)) ← 0 (Y) = 0 to 7	36
	SD	(D(Y)) ← 1 (Y) = 0 to 7	37
	SZD	(D(Y)) = 0 ? (Y) = 4 to 7	39
	OEA	(E_1, E_0) ← (A_1, A_0)	34
	IAE	(A_2–A_0) ← (E_2–E_0)	32
	OGA	(G) ← (A)	34
	IAG	(A) ← (G)	32
Carrier wave control operation	SCAR	(CAR) ← 1	37
	RCAR	(CAR) ← 0	35
Other operation	NOP	(PC) ← (PC) + 1	34
	POF	RAM back-up	34
	SNZP	(P) = 1 ?	38
	CCK	STCK changes to f(X_{IN})	31
	TLOA	(LO_1, LO_0) ← (A_1, A_0)	43
	URSC	(URS) ← 1	44
	TPU0A	($PU0_3$–$PU0_0$) ← (A_3–A_0)	43
	TPU1A	($PU1_3$–$PU1_0$) ← (A_3–A_0)	43
	WRST	(WDF1) ← 0	45

MACHINE INSTRUCTIONS (INDEX BY ALPHABET)

A n (Add n and accumulator)

Instruction code

D_8								D_0	
0	1	0	1	0	n_3	n_2	n_1	n_0	$_2$

0	A	n	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Overflow = 0

Operation: (A) ← (A) + n
n = 0 to 15

Grouping: Arithmetic operation

Description: Adds the value n in the immediate field to register A.
The contents of carry flag CY remains unchanged.
Skips the next instruction when there is no overflow as the result of operation.

AM (Add accumulator and Memory)

Instruction code

D_8								D_0	
0	0	0	0	0	1	0	1	0	$_2$

0	0	A	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (A) + (M(DP))

Grouping: Arithmetic operation

Description: Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.

AMC (Add accumulator, Memory and Carry)

Instruction code

D_8								D_0	
0	0	0	0	0	1	0	1	1	$_2$

0	0	B	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0/1	–

Operation: (A) ← (A) + (M(DP)) + (CY)
(CY) ← Carry

Grouping: Arithmetic operation

Description: Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.

B a (Branch to address a)

Instruction code

D_8								D_0	
1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$

1	8 +a	a	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PC_L) ← a_6–a_0

Grouping: Branch operation

Description: Branch within a page : Branches to address a in the identical page.

BA a (Branch to address a + Accumulator)

Instruction code

D8								D0	
0	0	0	0	0	0	0	0	1	$_2$
1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$

0	0	1	$_{16}$
1	8 +a	a	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation: $(PC_L) \leftarrow a_6$–a_4, A_3–A_0

Grouping: Branch operation

Description: Branch within a page : Branches to address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of the address a in the identical page with register A.

BL p, a (Branch Long to address a in page p)

Instruction code

D8								D0	
0	0	0	1	1	p_3	p_2	p_1	p_0	$_2$
1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$

0	3	p	$_{16}$
1	8 +a	a	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation: $(PC_H) \leftarrow (P)$
$(PC_L) \leftarrow a_6$–a_0

Grouping: Branch operation

Description: Branch out of a page : Branches to address a in page p.

Note: p is 0 to 15.

BLA p, a (Branch Long to address a in page p)

Instruction code

D8								D0	
0	0	0	0	1	0	0	0	0	$_2$
1	1	a_6	a_5	a_4	p_3	p_2	p_1	p_0	$_2$

0	1	0	$_{16}$
1	8 +a	p	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation: $(PC_H) \leftarrow (P)$
$(PC_L) \leftarrow (a_6$–a_4, A_3–$A_0)$

Grouping: Branch operation

Description: Branch within a page : Branches to address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of the address a in page p with register A.

Note: p is 0 to 15.

BM a (Branch and Mark to address a in page 2)

Instruction code

D8								D0	
1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$

1	a	a	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(SK(SP)) \leftarrow (PC)$
$(SP) \leftarrow (SP) + 1$
$(PC_H) \leftarrow 2$
$(PC_L) \leftarrow a_6$–a_0

Grouping: Subroutine call operation

Description: Call the subroutine in page 2 : Calls the subroutine at address a in page 2.

BML p, a (Branch and Mark Long to address a in page p)

Instruction code

D_8								D_0	
0	0	1	1	1	p_3	p_2	p_1	p_0	$_2$
1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$

0	7	p	$_{16}$
1	a	a	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation:
(SK(SP)) ← (PC)
(SP) ← (SP) + 1
(PC_H) ← p
(PC_L) ← a_6–a_0

Grouping: Subroutine call operation

Description: Call the subroutine : Calls the subroutine at address a in page p.

Note: p is 0 to 15.

BMLA p, a (Branch and Mark Long to address a in page p)

Instruction code

D_8								D_0	
0	0	1	0	1	0	0	0	0	$_2$
1	0	a_6	a_5	a_4	p_3	p_2	p_1	p_0	$_2$

0	5	0	$_{16}$
1	a	p	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation:
(SK(SP)) ← (PC)
(SP) ← (SP) + 1
(PC_H) ← p
(PC_L) ← (a_6–a_4, A_3–A_0)

Grouping: Subroutine call operation

Description: Call the subroutine : Calls the subroutine at address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of address a in page p with register A.

Note: p is 0 to 15.

CCK (Change system Clock to f(X_{IN}))

Instruction code

D_8								D_0	
0	0	1	0	1	1	0	0	1	$_2$

0	5	9	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: Change to STCK = f(X_{IN})

Grouping: Other operation

Description: Changes system clock (STCK) from f(X_{IN})/8 to f(X_{IN}). Execute this instruction at address 0 in page 0.

CLD (CLear port D)

Instruction code

D_8								D_0	
0	0	0	0	1	0	0	0	1	$_2$

0	1	1	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D) ← 1

Grouping: Input/Output operation

Description: Clears (0) to port D (high-impedance state).

CMA (CoMplement of Accumulator)

Instruction code

D_8								D_0	
0	0	0	0	1	1	1	0	0	$_2$

0	1	C	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← $\overline{(A)}$

Grouping: Arithmetic operation

Description: Stores the one's complement for register A's contents in register A.

DEY (DEcrement register Y)

Instruction code

D_8								D_0	
0	0	0	0	1	0	1	1	1	$_2$

0	1	7	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 15

Operation: (Y) ← (Y) – 1

Grouping: RAM addresses

Description: Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.

IAE (Input Accumulator from port E)

Instruction code

D_8								D_0	
0	0	1	0	1	0	1	1	0	$_2$

0	5	6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A_2–A_0) ← (E_2–E_0)

Grouping: Input/Output operation

Description: Transfers the contents of port E to register A.

IAG (Input Accumulator from port G)

Instruction code

D_8								D_0	
0	0	0	1	0	1	0	0	0	$_2$

0	2	8	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (G)

Grouping: Input/Output operation

Description: Transfers the contents of port G to register A.

INY (INcrement register Y)

Instruction code

D_8								D_0	
0	0	0	0	1	0	0	1	1	$_2$

0	1	3	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 0

Operation: (Y) ← (Y) + 1

Grouping: RAM addresses

Description: Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.

LA n (Load n in Accumulator)

Instruction code

D_8								D_0	
0	1	0	1	1	n_3	n_2	n_1	n_0	$_2$

0	B	n	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Continuous description

Operation: (A) ← n
n = 0 to 15

Grouping: Arithmetic operation

Description: Loads the value n in the immediate field to register A.
When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.

LGOP (LoGic OPeration between accumulator and register E)

Instruction code

D_8								D_0	
0	0	1	0	0	0	0	0	1	$_2$

0	4	1	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: Logic operation XOR, OR, AND

Grouping: Arithmetic operation

Description: Executes the logic operation selected by logic operation selection register LO between the contents of register A and register E, and stores the result in register A.

LXY x, y (Load register X and Y with x and y)

Instruction code

D_8								D_0	
0	1	1	x_1	x_0	y_3	y_2	y_1	y_0	$_2$

0	C +x	y	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Continuous description

Operation: (X) ← x, x = 0 to 3
(Y) ← y, y = 0 to 15

Grouping: RAM addresses

Description: Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.

NOP (No OPeration)

Instruction code

D8								D0		
0	0	0	0	0	0	0	0	0	0	0	0
$_2$									$_{16}$	

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PC) ← (PC) + 1

Grouping: Other operation

Description: No operation

OEA (Output port E from Accumulator)

Instruction code

D8								D0		
0	1	0	0	0	0	1	0	0	8	4
$_2$									$_{16}$	

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (E_1, E_0) ← (A_1, A_0)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port E.

OGA (Output port G from Accumulator)

Instruction code

D8								D0		
0	1	0	0	0	0	0	0	0	8	0
$_2$									$_{16}$	

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (G) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port G.

POF (Power OFf1)

Instruction code

D8								D0		
0	0	0	0	0	1	1	0	1	0	0	D
$_2$									$_{16}$	

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: RAM back-up

Grouping: Other operation

Description: Puts the system in RAM back-up state.

RAR (Rotate Accumulator Right)

Instruction code

D_8								D_0	
0	0	0	0	1	1	1	0	1	$_2$

0	1	D	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0/1	–

Operation: → CY → $A_3A_2A_1A_0$

Grouping: Arithmetic operation

Description: Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.

RB j (Reset Bit)

Instruction code

D_8								D_0	
0	0	1	0	0	1	1	j_1	j_0	$_2$

0	4	C +j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Mj(DP)) ← 0
j = 0 to 3

Grouping: Bit operation

Description: Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).

RC (Reset Carry flag)

Instruction code

D_8								D_0	
0	0	0	0	0	0	1	1	0	$_2$

0	0	6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0	–

Operation: (CY) ← 0

Grouping: Arithmetic operation

Description: Clears (0) to carry flag CY.

RCAR (Reset CAR flag)

Instruction code

D_8								D_0	
0	1	0	0	0	0	1	1	0	$_2$

0	8	6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (CAR) ← 0

Grouping: Carrier wave control operation

Description: Clears (0) to port CARR output flag.

RD (Reset port D specified by register Y)

Instruction code

D8								D0	
0	0	0	0	1	0	1	0	0	$_2$

0	1	4	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D(Y)) ← 0
However,
(Y) = 0 to 7

Grouping: Input/Output operation

Description: Clears (0) to a bit of port D specified by register Y (high-impedance state).

RT (ReTurn from subroutine)

Instruction code

D8								D0	
0	0	1	0	0	0	1	0	0	$_2$

0	4	4	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	2	–	–

Operation: (SP) ← (SP) – 1
(PC) ← (SK(SP))

Grouping: Return operation

Description: Returns from subroutine to the routine called the subroutine.

RTS (ReTurn form subroutine and Skip)

Instruction code

D8								D0	
0	0	1	0	0	0	1	0	1	$_2$

0	4	5	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	2	–	Skip at uncondition

Operation: (SP) ← (SP) – 1
(PC) ← (SK(SP))

Grouping: Return operation

Description: Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.

SB j (Set Bit)

Instruction code

D8								D0	
0	0	1	0	1	1	1	j_1	j_0	$_2$

0	5	C +j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Mj(DP)) ← 0
j = 0 to 3

Grouping: Bit operation

Description: Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).

SC (Set Carry flag)

Instruction code

D8								D0		
0	0	0	0	0	0	1	1	1	0	0	7
$_2$									$_{16}$		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	1	–

Operation: (CY) ← 1

Grouping: Arithmetic operation

Description: Sets (1) to carry flag CY.

SCAR (Set CAR flag)

Instruction code

D8								D0		
0	1	0	0	0	0	1	1	1	0	8	7
$_2$									$_{16}$		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (CAR) ← 1

Grouping: Carrier wave control operation

Description: Sets (1) to port CARR output flag (CAR).

SD (Set port D specified by register Y)

Instruction code

D8								D0		
0	0	0	0	1	0	1	0	1	0	1	5
$_2$									$_{16}$		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D(Y)) ← 1
(Y) = 0 to 7

Grouping: Input/Output operation

Description: Sets (1) to a bit of port D specified by register Y.

SEA n (Skip Equal, Accumulator with immediate data n)

Instruction code

D8								D0		
0	0	0	1	0	0	1	0	1	0	2	5
$_2$									$_{16}$		
0	1	0	1	1	n_3	n_2	n_1	n_0	0	B	n
$_2$									$_{16}$		

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	(A) = n, n = 0 to 15

Operation: (A) = n ?
n = 0 to 15

Grouping: Comparison operation

Description: Skips the next instruction when the contents of register A is equal to the value n in the immediate field.

SEAM (Skip Equal, Accumulator with Memory)

Instruction code

D8								D0			
0	0	0	1	0	0	1	1	0	$_2$	0 2 6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(A) = (M(DP))

Operation: (A) = (M(DP)) ?

Grouping: Comparison operation

Description: Skips the next instruction when the contents of register A is equal to the contents of M(DP).

SNZP (Skip if Non Zero condition of Power down flag)

Instruction code

D8								D0			
0	0	0	0	0	0	0	1	1	$_2$	0 0 3	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(P) = 1

Operation: (P) = 1 ?

Grouping: Other operation

Description: Skips the next instruction when P flag is “1”. After skipping, P flag remains unchanged.

SNZT1 (Skip if Non Zero condition of Timer 1 underflow flag)

Instruction code

D8								D0			
0	0	1	0	0	0	0	1	0	$_2$	0 4 2	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(T1F) = 1

Operation: (T1F) = 1 ?
(T1F) ← 0

Grouping: Timer operation

Description: Clears T1F flag and skips the next instruction when the contents of T1F flag is “1.”

SNZT2 (Skip if Non Zero condition of Timer 2 inerrupt request flag)

Instruction code

D8								D0			
0	0	1	0	1	0	0	1	0	$_2$	0 5 2	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(T2F) = 1

Operation: (T2F) = 1 ?
(T2F) ← 0

Grouping: Timer operation

Description: Clears T2F flag and skips the next instruction when the contents of T2F flag is “1.”

SZB j (Skip if Zero, Bit)

Instruction code

D8								D0		
0	0	0	1	0	0	0	j_1	j_0	$_2$	0 2 j $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Mj(DP)) = 0 j = 0 to 3

Operation: (Mj(DP)) = 0 ?
j = 0 to 3

Grouping: Bit operation

Description: Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is “0.”

SZC (Skip if Zero, Carry flag)

Instruction code

D8								D0		
0	0	0	1	0	1	1	1	1	$_2$	0 2 F $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(CY) = 0

Operation: (CY) = 0 ?

Grouping: Arithmetic operation

Description: Skips the next instruction when the contents of carry flag CY is “0.”

SZD (Skip if Zero, port D specified by register Y)

Instruction code

D8								D0		
0	0	0	1	0	0	1	0	0	$_2$	0 2 4 $_{16}$
0	0	0	1	0	1	0	1	1	$_2$	0 2 B $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	(D(Y)) = 0 (Y) = 4 to 7

Operation: (D(Y)) = 0 ?
(Y) = 4 to 7

Grouping: Input/Output operation

Description: Skips the next instruction when a bit of port D specified by register Y is “0.”

T1AB (Transfer data to timer 1 and register R1 from Accumulator and register B)

Instruction code

D8								D0		
0	0	1	0	0	0	1	1	1	$_2$	0 4 7 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: at timer 1 stop ($V1_0$=0)
($R1_7$–$R1_4$) ← (B), ($R1_3$–$R1_0$) ← (A)
($T1_7$–$T1_4$) ← (B), ($T1_3$–$T1_0$) ← (A)
at timer 1 operating ($V1_0$=1)
($R1_7$–$R1_4$) ← (B), ($R1_3$–$R1_0$) ← (A)

Grouping: Timer operation

Description: At timer 1 stop ($V1_0$ = 0), transfers the contents of register A and register B to timer 1 and reload register R1.
At timer 1 operating ($V1_0$ = 1), transfers the contents of register A and register B to reload register R1.

T2AB (Transfer data to timer 2 and register R2L from Accumulator and register B)

Instruction code

D_8								D_0	
0	1	0	0	0	1	0	0	0	$_2$

0	8	8	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(R2L_7–R2L_4) \leftarrow (B)$
$(R2L_3–R2L_0) \leftarrow (A)$
$(T2_7–T2_4) \leftarrow (B)$
$(T2_3–T2_0) \leftarrow (A)$

Grouping: Timer operation

Description: Transfers the contents of registers A and B to timer 2 and timer 2 reload register R2L.

T2HAB (Transfer data to register R2H Accumulator from register B)

Instruction code

D_8								D_0	
0	1	0	0	0	1	0	0	1	$_2$

0	8	9	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(R2H_7–R2H_4) \leftarrow (B)$
$(R2H_3–R2H_0) \leftarrow (A)$

Grouping: Timer operation

Description: Transfers the contents of register A and register B to reload register R2H.

T2R2L (Transfer data to timer 2 from register R2L)

Instruction code

D_8								D_0	
0	0	1	0	1	0	0	1	1	$_2$

0	5	3	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(T2_7–T2_4) \leftarrow (R2L_7–R2L_4)$
$(T2_3–T2_0) \leftarrow (R2L_3–R2L_0)$

Grouping: Timer operation

Description: Transfers the contents of reload register R2L to timer 2.

TAB (Transfer data to Accumulator from register B)

Instruction code

D_8								D_0	
0	0	0	0	1	1	1	1	0	$_2$

0	1	E	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(A) \leftarrow (B)$

Grouping: Register to register transfer

Description: Transfers the contents of register B to register A.

TAB1 (Transfer data to Accumulator and register B from timer 1)

Instruction code

D_8								D_0	
0	0	1	0	1	0	1	1	1	$_2$

0	5	7	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (B) ← ($T1_7$–$T1_4$)
(A) ← ($T1_3$–$T1_0$)

Grouping: Timer operation

Description: Transfers the contents of timer 1 to registers A and B.

TAB2 (Transfer data to Accumulator and register B from timer 2)

Instruction code

D_8								D_0	
0	0	1	0	0	0	0	0	0	$_2$

0	4	0	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (B) ← ($T2_7$–$T2_4$)
(A) ← ($T2_3$–$T2_0$)

Grouping: Timer operation

Description: Transfers the contents of timer 2 to registers A and B.

TABE (Transfer data to Accumulator and register B from register E)

Instruction code

D_8								D_0	
0	0	0	1	0	1	0	1	0	$_2$

0	2	A	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (B) ← (ER_7–ER_4)
(A) ← (ER_3–ER_0)

Grouping: Register to register transfer

Description: Transfers the contents of register E to registers A and B.

TABP p (Transfer data to Accumulator and register B from Program memory in page p)

Instruction code

D_8								D_0	
0	1	0	0	1	p_3	p_2	p_1	p_0	$_2$

0	9	p	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	3	– 0/1	–

Operation: SK(SP)) ← (PC) , (SP) ← (SP) + 1
(PC_H) ← p, p = 0 to 7, (PC_L) ← (DR_2–DR_0, A_3–A_0)
When URS = 0,
(B) ← (ROM(PC))$_{7\ to\ 4}$, (A) ← (ROM(PC))$_{3\ to\ 0}$
When URS = 1,
(CY) ← (ROM(PC))$_8$
(B) ← (ROM(PC))$_{7\ to\ 4}$, (A) ← (ROM(PC))$_{3\ to\ 0}$
(SP) ← (SP) – 1, (PC) ← (SK(SP))

Note: p is 0 to 15.

Grouping: Arithmetic operation

Description:
Transfers bits 7 to 4 to register B and bits 3 to 0 to register A when URS flag is cleared to “0.” These bits 7 to 0 are the ROM pattern in address (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0) specified by registers A and D in page p.
Transfers bit 8 of ROM pattern is transferred to flag CY when URS flag is set to “1” (after the URSC instruction is executed).
(One of stack is used when the TABP p instruction is executed.)

TAM j (Transfer data to Accumulator from Memory)

Instruction code

D_8								D_0	
0	0	1	1	0	0	1	j_1	j_0	$_2$

0	6	4+j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 3

Grouping: RAM to register transfer

Description: After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

TAY (Transfer data to Accumulator from register Y)

Instruction code

D_8								D_0	
0	0	0	0	1	1	1	1	1	$_2$

0	1	F	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (Y)

Grouping: Register to register transfer

Description: Transfers the contents of register Y to register A.

TBA (Transfer data to register B from Accumulator)

Instruction code

D_8								D_0	
0	0	0	0	0	1	1	1	0	$_2$

0	0	E	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (B) ← (A)

Grouping: Register to register transfer

Description: Transfers the contents of register A to register B.

TDA (Transfer data to register D from Accumulator)

Instruction code

D_8								D_0	
0	0	0	1	0	1	0	0	1	$_2$

0	2	9	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (DR_2–DR_0) ← (A_2–A_0)

Grouping: Register to register transfer

Description: Transfers the contents of register A to register D.

TEAB (Transfer data to register E from Accumulator and register B)

Instruction code

D_8								D_0	
0	0	0	0	1	1	0	1	0	$_2$

0	1	A	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (ER_7–ER_4) ← (B)
(ER_3–ER_0) ← (A)

Grouping: Register to register transfer

Description: Transfers the contents of register A and register B to register E.

TLOA (Transfer data to register LO from Accumulator)

Instruction code

D_8								D_0	
0	0	1	0	1	1	0	0	0	$_2$

0	5	8	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (LO_1, LO_0) ← (A_1, A_0)

Grouping: Other operation

Description: Transfers the contents of register A to logic operation selection register LO.

TPU0A (Transfer data to register PU0 from Accumulator)

Instruction code

D_8								D_0	
0	1	0	0	0	1	1	1	1	$_2$

0	8	F	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: ($PU0_3$–$PU0_0$) ← (A_3–A_0)

Grouping: Other operation

Description: Transfers the contents of register A to pull-up control register PU0.

TPU1A (Transfer data to register PU1 from Accumulator)

Instruction code

D_8								D_0	
0	1	0	0	0	1	1	1	0	$_2$

0	8	E	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: ($PU1_3$–$PU1_0$) ← (A_3–A_0)

Grouping: Other operation

Description: Transfers the contents of register A to pull-up control register PU1.

TV1A (Transfer data to register V1 from Accumulator)

Instruction code

D_8								D_0	
0	0	1	0	1	0	1	1	1	$_2$

0	5	B	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: ($V1_2$–$V1_0$) ← (A_2–A_0)

Grouping: Timer operation

Description: Transfers the contents of register A to register V1.

TV2A (Transfer data to register V2 from Accumulator)

Instruction code

D_8								D_0	
0	0	1	0	1	1	0	1	0	$_2$

0	5	A	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: ($V2_3$–$V2_0$) ← (A_3–A_0)

Grouping: Timer operation

Description: Transfers the contents of register A to register V2.

TYA (Transfer data to regiser Y from Accumulator)

Instruction code

D_8								D_0	
0	0	0	0	0	1	1	0	0	$_2$

0	0	C	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Y) ← (A)

Grouping: Register to register transfer

Description: Transfers the contents of register A to register Y.

URSC (Sets Upper ROM Code reference enable flag)

Instruction code

D_8								D_0	
0	1	0	0	0	0	0	1	0	$_2$

0	8	2	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (URS) ← 1

Grouping: Other operation

Description: Sets the most significant ROM code reference enable flag (URS) to “1.”

WRST (Watchdog timer ReSeT)

Instruction code

D8								D0	
0	0	0	0	0	1	1	1	1	$_2$

0	0	F	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (WDF1) ← 0

Grouping: Other operation

Description: Initializes the watchdog timer flag (WDF1).

XAM j (eXchange Accumulator and Memory data)

Instruction code

D8								D0	
0	0	1	1	0	0	0	j_1	j_0	$_2$

0	6	j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 3

Grouping: RAM to register transfer

Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

XAMD j (eXchange Accumulator and Memory data and Decrement register Y and skip)

Instruction code

D8								D0	
0	0	1	1	0	1	1	j_1	j_0	$_2$

0	6	C +j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 15

Operation: (A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 3
(Y) ← (Y) – 1

Grouping: RAM to register transfer

Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.

XAMI j (eXchange Accumulator and Memory data and Increment register Y and skip)

Instruction code

D8								D0	
0	0	1	1	0	1	0	j_1	j_0	$_2$

0	6	8 +j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 0

Operation: (A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 3
(Y) ← (Y) + 1

Grouping: RAM to register transfer

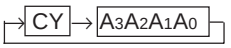
Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.

MACHINE INSTRUCTIONS (INDEX BY FUNCTION)

Type of instructions	Mnemonic	Instruction code D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation	Number of words	Number of cycles	Function
Register to register transfer	TAB	0	0	0	0	1	1	1	1	0	0 1 E	1	1	(A) ← (B)
	TBA	0	0	0	0	0	1	1	1	0	0 0 E	1	1	(B) ← (A)
	TAY	0	0	0	0	1	1	1	1	1	0 1 F	1	1	(A) ← (Y)
	TYA	0	0	0	0	0	1	1	0	0	0 0 C	1	1	(Y) ← (A)
	TEAB	0	0	0	0	1	1	0	1	0	0 1 A	1	1	(ER_7–ER_4) ← (B) (ER_3–ER_0) ← (A)
	TABE	0	0	0	1	0	1	0	1	0	0 2 A	1	1	(B) ← (ER_7–ER_4) (A) ← (ER_3–ER_0)
	TDA	0	0	0	1	0	1	0	0	1	0 2 9	1	1	(DR_2–DR_0) ← (A_2–A_0)
RAM addresses	LXY x, y	0	1	1	x_1	x_0	y_3	y_2	y_1	y_0	0 C +x y	1	1	(X) ← x, x = 0 to 3 (Y) ← y, y = 0 to 15
	INY	0	0	0	0	1	0	0	1	1	0 1 3	1	1	(Y) ← (Y) + 1
	DEY	0	0	0	0	1	0	1	1	1	0 1 7	1	1	(Y) ← (Y) – 1
RAM to register transfer	TAM j	0	0	1	1	0	0	1	j_1	j_0	0 6 4 +j	1	1	(A) ← (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3
	XAM j	0	0	1	1	0	0	0	j_1	j_0	0 6 j	1	1	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3
	XAMD j	0	0	1	1	0	1	1	j_1	j_0	0 6 C +j	1	1	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) – 1
	XAMI j	0	0	1	1	0	1	0	j_1	j_0	0 6 8 +j	1	1	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) + 1

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the contents of register B to register A.
–	–	Transfers the contents of register A to register B.
–	–	Transfers the contents of register Y to register A.
–	–	Transfers the contents of register A to register Y.
–	–	Transfers the contents of registers A and B to register E.
–	–	Transfers the contents of register E to registers A and B.
–	–	Transfers the contents of register A to register D.
Continuous description	–	Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.
(Y) = 0	–	Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.
(Y) = 15	–	Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.
–	–	After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
–	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
(Y) = 15	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.
(Y) = 0	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.

MACHINE INSTRUCTIONS (CONTINUED)

Type of instructions	Mnemonic	Instruction code									Hexadecimal notation			Number of words	Number of cycles	Function
		D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0						
Arithmetic operation	LA n	0	1	0	1	1	n_3	n_2	n_1	n_0	0	B	n	1	1	(A) ← n n = 0 to 15
	TABP p	0	1	0	0	1	p_3	p_2	p_1	p_0	0	9	p	1	3	(SK(SP)) ← (PC) (SP) ← (SP) + 1 (PC_H) ← p (Note) (PC_L) ← (DR_2–DR_0, A_3–A_0) When URS=0, (B) ← (ROM(PC))$_{7\ to\ 4}$ (A) ← (ROM(PC))$_{3\ to\ 0}$ When URS=1, (CY) ← (ROM(PC))$_8$ (B) ← (ROM(PC))$_{7\ to\ 4}$ (A) ← (ROM(PC))$_{3\ to\ 0}$ (SP) ← (SP) – 1 (PC) ← (SK(SP))
	AM	0	0	0	0	0	1	0	1	0	0	0	A	1	1	(A) ← (A) + (M(DP))
	AMC	0	0	0	0	0	1	0	1	1	0	0	B	1	1	(A) ← (A) + (M(DP))+ (CY) (CY) ← Carry
	A n	0	1	0	1	0	n_3	n_2	n_1	n_0	0	A	n	1	1	(A) ← (A) + n n = 0 to 15
	SC	0	0	0	0	0	0	1	1	1	0	0	7	1	1	(CY) ← 1
	RC	0	0	0	0	0	0	1	1	0	0	0	6	1	1	(CY) ← 0
	SZC	0	0	0	1	0	1	1	1	1	0	2	F	1	1	(CY) = 0 ?
	CMA	0	0	0	0	1	1	1	0	0	0	1	C	1	1	(A) ← ($\overline{A}$)
	RAR	0	0	0	0	1	1	1	0	1	0	1	D	1	1	
	LGOP	0	0	1	0	0	0	0	0	1	0	4	1	1	1	Logic operation instruction XOR, OR, AND

Note: p is 0 to 15.

Skip condition	Carry flag CY	Detailed description
Continuous description	–	Loads the value n in the immediate field to register A. When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.
–	–	Transfers bits 7 to 4 to register B and bits 3 to 0 to register A when URS flag is cleared to “0.” These bits 7 to 0 are the ROM pattern in address (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0) specified by registers A and D in page p.
	0/1	Transfers bit 8 of ROM pattern is transferred to flag CY when URS flag is set to “1” (after the URSC instruction is executed). (One of stack is used when the TABP p instruction is executed.)
–	–	Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.
–	0/1	Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.
Overflow = 0	–	Adds the value n in the immediate field to register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation.
–	1	Sets (1) to carry flag CY.
–	0	Clears (0) to carry flag CY.
(CY) = 0	–	Skips the next instruction when the contents of carry flag CY is “0.”
–	–	Stores the one‘s complement for register A‘s contents in register A.
–	0/1	Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.
–	–	Executes the logic operation selected by logic operation selection register LO between the contents of register A and register E, and stores the result in register A.

MACHINE INSTRUCTIONS (CONTINUED)

Type of instructions \ Parameter	Mnemonic	Instruction code D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation	Number of words	Number of cycles	Function
Bit operation	SB j	0	0	1	0	1	1	1	j_1	j_0	0 5 C+j	1	1	(Mj(DP)) ← 1 j = 0 to 3
	RB j	0	0	1	0	0	1	1	j_1	j_0	0 4 C+j	1	1	(Mj(DP)) ← 0 j = 0 to 3
	SZB j	0	0	0	1	0	0	0	j_1	j_0	0 2 j	1	1	(Mj(DP)) = 0 ? j = 0 to 3
Comparison operation	SEAM	0	0	0	1	0	0	1	1	0	0 2 6	1	1	(A) = (M(DP)) ?
	SEA n	0	0	0	1	0	0	1	0	1	0 2 5	2	2	(A) = n ? n = 0 to 15
		0	1	0	1	1	n_3	n_2	n_1	n_0	0 B n			
Branch operation	B a	1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8+a a	1	1	(PC_L) ← a_6–a_0
	BL p, a	0	0	0	1	1	p_3	p_2	p_1	p_0	0 3 p	2	2	(PC_H) ← p (PC_L) ← a_6–a_0 (Note)
		1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8+a a			
	BA a	0	0	0	0	0	0	0	0	1	0 0 1	2	2	(PC_L) ← (a_6–a_4, A_3–A_0)
		1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8+a a			
	BLA p, a	0	0	0	0	1	0	0	0	0	0 1 0	2	2	(PC_H) ← p (PC_L) ← (a_6–a_4, A_3–A_0)
		1	1	a_6	a_5	a_4	p_3	p_2	p_1	p_0	1 8+a p			(Note)

Note: p is 0 to 15.

Skip condition	Carry flag CY	Detailed description
–	–	Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
–	–	Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
(Mj(DP)) = 0 j = 0 to 3	–	Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is “0.”
(A) = (M(DP))	–	Skips the next instruction when the contents of register A is equal to the contents of M(DP).
(A) = n n = 0 to 15	–	Skips the next instruction when the contents of register A is equal to the value n in the immediate field.
–	–	Branch within a page : Branches to address a in the identical page.
–	–	Branch out of a page : Branches to address a in page p.
–	–	Branch within a page : Branches to address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of the address a in the identical page with register A.
–	–	Branch out of a page : Branches to address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of the address a in page p with register A.

MACHINE INSTRUCTIONS (CONTINUED)

Parameter / Type of instructions	Mnemonic	Instruction code									Hexadecimal notation			Number of words	Number of cycles	Function
		D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0						
Subroutine operation	BM a	1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1	a	a	1	1	(SK(SP)) ← (PC) (SP) ← (SP) + 1 (PC_H) ← 2 (PC_L) ← a_6–a_0
	BML p, a	0	0	1	1	1	p_3	p_2	p_1	p_0	0	7	p	2	2	(SK(SP)) ← (PC) (SP) ← (SP) + 1 (PC_H) ← p
		1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1	a	a			(PC_L) ← a_6–a_0 (Note)
	BMLA p, a	0	0	1	0	1	0	0	0	0	0	5	0	2	2	(SK(SP)) ← (PC) (SP) ← (SP) + 1
		1	0	a_6	a_5	a_4	p_3	p_2	p_1	p_0	1	a	p			(PC_H) ← p (PC_L) ← (a_6–a_4, A_3–A_0) (Note)
Return operation	RT	0	0	1	0	0	0	1	0	0	0	4	4	1	2	(SP) ← (SP) – 1 (PC) ← (SK(SP))
	RTS	0	0	1	0	0	0	1	0	1	0	4	5	1	2	(SP) ← (SP) – 1 (PC) ← (SK(SP))
Timer operation	T1AB	0	0	1	0	0	0	1	1	1	0	4	7	1	1	at timer 1 stop ($V1_0$=0) ($R1_7$–$R1_4$) ← (B), ($R1_3$–$R1_0$) ← (A) ($T1_7$–$T1_4$) ← (B), ($T1_3$–$T1_0$) ← (A) at timer 1 operating ($V1_0$=1) ($R1_7$–$R1_4$) ← (B), ($R1_3$–$R1_0$) ← (A)
	TAB1	0	0	1	0	1	0	1	1	1	0	5	7	1	1	(B) ← ($T1_7$–$T1_4$) (A) ← ($T1_3$–$T1_0$)
	TV1A	0	0	1	0	1	1	0	1	1	0	5	B	1	1	($V1_2$–$V1_0$) ← (A_2–A_0)
	SNZT1	0	0	1	0	0	0	0	1	0	0	4	2	1	1	(T1F) = 1 ? (T1F) ← 0
	T2AB	0	1	0	0	0	1	0	0	0	0	8	8	1	1	($R2L_7$–$R2L_4$) ← (B) ($R2L_3$–$R2L_0$) ← (A) ($T2_7$–$T2_4$) ← (B), ($T2_3$–$T2_0$) ← (A)

Note : p is 0 to 15.

Skip condition	Carry flag CY	Detailed description
–	–	Call the subroutine in page 2 : Calls the subroutine at address a in page 2.
–	–	Call the subroutine : Calls the subroutine at address a in page p.
–	–	Call the subroutine : Calls the subroutine at address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of address a in page p with register A.
–	–	Returns from subroutine to the routine called the subroutine.
Skip at uncondition	–	Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.
–	–	At timer 1 stop ($V1_0$ = 0), transfers the contents of register A and register B to timer 1 and reload register R1. At timer 1 operating ($V1_0$ = 1), transfers the contents of register A and register B to reload register R1.
–	–	Transfers the contents of timer 1 to registers A and B.
–	–	Transfers the contents of register A to registers V1.
(T1F) = 1	–	Clears T1F flag and skips the next instruction when the contents of T1F flag is “1.”
–	–	Transfers the contents of register A and register B to timer 2 and reload register R2L.

MACHINE INSTRUCTIONS (CONTINUED)

Parameter / Type of instructions	Mnemonic	Instruction code										Number of words	Number of cycles	Function
		D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
Timer operation	TAB2	0	0	1	0	0	0	0	0	0	0 4 0	1	1	(B) ← ($T2_7$–$T2_4$), (A) ← ($T2_3$–$T2_0$)
	TV2A	0	0	1	0	1	1	0	1	0	0 5 A	1	1	($V2_3$–$V2_0$) ← (A_3–A_0)
	SNZT2	0	0	1	0	1	0	0	1	0	0 5 2	1	1	(T2F) = 1 ? (T2F) ← 0
	T2HAB	0	1	0	0	0	1	0	0	1	0 8 9	1	1	($R2H_7$–$R2H_4$) ← (B) ($R2H_3$–$R2H_0$) ← (A)
	T2R2L	0	0	1	0	1	0	0	1	1	0 5 3	1	1	($T2_7$–$T2_4$) ← ($R2L_7$–$R2L_4$) ($T2_3$–$T2_0$) ← ($R2L_3$–$R2L_0$)
Carrier wave control operation	SCAR	0	1	0	0	0	0	1	1	1	0 8 7	1	1	(CAR) ← 1
	RCAR	0	1	0	0	0	0	1	1	0	0 8 6	1	1	(CAR) ← 0
Input/Output operation	CLD	0	0	0	0	1	0	0	0	1	0 1 1	1	1	(D) ← 0
	RD	0	0	0	0	1	0	1	0	0	0 1 4	1	1	(D(Y)) ← 0 (Y) = 0 to 7
	SD	0	0	0	0	1	0	1	0	1	0 1 5	1	1	(D(Y)) ← 1 (Y) = 0 to 7
	SZD	0	0	0	1	0	0	1	0	0	0 2 4	2	2	(D(Y)) = 0 ? (Y) = 4 to 7
		0	0	0	1	0	1	0	1	1	0 2 B			
	OEA	0	1	0	0	0	0	1	0	0	0 8 4	1	1	(E_1, E_0) ← (A_1, A_0)
	IAE	0	0	1	0	1	0	1	1	0	0 5 6	1	1	(A_2–A_0) ← (E_2–E_0)
	OGA	0	1	0	0	0	0	0	0	0	0 8 0	1	1	(G) ← (A)
	IAG	0	0	0	1	0	1	0	0	0	0 2 8	1	1	(A) ← (G)

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the contents of timer 2 to registers A and B.
–	–	Transfers the contents of register A to registers V2.
(T2F) = 1	–	Clears T2F flag and skips the next instruction when the contents of T2F flag is “1.”
–	–	Transfers the contents of register A and register B to reload register R2H.
–	–	Transfers the contents of reload register R2L to timer 2.
–	–	Sets (1) to port CARR output flag (CAR).
–	–	Clears (0) to port CARR output flag (CAR).
–	–	Clears (0) to port D (high-impedance state).
–	–	Clears (0) to a bit of port D specified by register Y (high-impedance state).
–	–	Sets (1) to a bit of port D specified by register Y.
(D(Y)) = 0 (Y) = 4 to 7	–	Skips the next instruction when a bit of port D specified by register Y is “0.”
–	–	Outputs the contents of register A to port E.
–	–	Transfers the contents of port E to register A.
–	–	Outputs the contents of register A to port G.
–	–	Transfers the contents of port G to register A.

Parameter / Type of instructions	Mnemonic	Instruction code										Number of words	Number of cycles	Function
		D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
Other operation	NOP	0	0	0	0	0	0	0	0	0	0 0 0	1	1	(PC) ← (PC) + 1
	POF	0	0	0	0	0	1	1	0	1	0 0 D	1	1	RAM back-up
	SNZP	0	0	0	0	0	0	0	1	1	0 0 3	1	1	(P) = 1 ?
	CCK	0	0	1	0	1	1	0	0	1	0 5 9	1	1	STCK changes to f(X_{IN})
	TLOA	0	0	1	0	1	1	0	0	0	0 5 8	1	1	(LO_1, LO_0) ← (A_1, A_0)
	URSC	0	1	0	0	0	0	0	1	0	0 8 2	1	1	(URS) ← 1
	TPU0A	0	1	0	0	0	1	1	1	1	0 8 F	1	1	($PU0_3$–$PU0_0$) ← (A_3–A_0)
	TPU1A	0	1	0	0	0	1	1	1	0	0 8 E	1	1	($PU1_3$–$PU1_0$) ← (A_3–A_0)
	WRST	0	0	0	0	0	1	1	1	1	0 0 F	1	1	(WDF1) ← 0

Skip condition	Carry flag CY	Detailed description
–	–	No operation
–	–	Puts the system in RAM back-up state.
(P) = 1	–	Skips the next instruction when P flag is “1.” After skipping, P flag remains unchanged.
–	–	System clock (STCK) changes to f(X_{IN}) from f(X_{IN})/8. Execute this CCK instruction at address 0 in page 0.
–	–	Transfers the contents of register A to the logic operation selection register LO.
–	–	Sets the most significant ROM code reference enable flag (URS) to “1.”
–	–	Transfers the contents of register A to register PU0.
–	–	Transfers the contents of register A to register PU1.
–	–	Initializes the watchdog timer flag (WDF1).

INSTRUCTION CODE TABLE

D_3–D_0 Hex. notation \ D_8–D_4	00000	00001	00010	00011	00100	00101	00110	00111	01000	01001	01010	01011	01100	01101	01110	01111	10000 10111	11000 11111
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10–17	18–1F
0000	0	NOP	BLA	SZB 0	BL	TAB2	BMLA	XAM 0	BML	OGA	TABP 0	A 0	LA 0	LXY 0,0	LXY 1,0	LXY 2,0	LXY 3,0	BM	B
0001	1	BA	CLD	SZB 1	BL	LGOP	—	XAM 1	BML	—	TABP 1	A 1	LA 1	LXY 0,1	LXY 1,1	LXY 2,1	LXY 3,1	BM	B
0010	2	—	—	SZB 2	BL	SNZT1	SNZT2	XAM 2	BML	URSC	TABP 2	A 2	LA 2	LXY 0,2	LXY 1,2	LXY 2,2	LXY 3,2	BM	B
0011	3	SNZP	INY	SZB 3	BL	—	T2R2L	XAM 3	BML	—	TABP 3	A 3	LA 3	LXY 0,3	LXY 1,3	LXY 2,3	LXY 3,3	BM	B
0100	4	—	RD	SZD	BL	RT	—	TAM 0	BML	OEA	TABP 4	A 4	LA 4	LXY 0,4	LXY 1,4	LXY 2,4	LXY 3,4	BM	B
0101	5	—	SD	SEAn	BL	RTS	—	TAM 1	BML	—	TABP 5	A 5	LA 5	LXY 0,5	LXY 1,5	LXY 2,5	LXY 3,5	BM	B
0110	6	RC	—	SEAM	BL	—	IAE	TAM 2	BML	RCAR	TABP 6	A 6	LA 6	LXY 0,6	LXY 1,6	LXY 2,6	LXY 3,6	BM	B
0111	7	SC	DEY	—	BL	T1AB	TAB1	TAM 3	BML	SCAR	TABP 7	A 7	LA 7	LXY 0,7	LXY 1,7	LXY 2,7	LXY 3,7	BM	B
1000	8	—	—	IAG	BL	—	TLOA	XAMI 0	BML	T2AB	TABP 8	A 8	LA 8	LXY 0,8	LXY 1,8	LXY 2,8	LXY 3,8	BM	B
1001	9	—	—	TDA	BL	—	CCK	XAMI 1	BML	T2HAB	TABP 9	A 9	LA 9	LXY 0,9	LXY 1,9	LXY 2,9	LXY 3,9	BM	B
1010	A	AM	TEAB	TABE	BL	—	TV2A	XAMI 2	BML	—	TABP 10	A 10	LA 10	LXY 0,10	LXY 1,10	LXY 2,10	LXY 3,10	BM	B
1011	B	AMC	—	—	BL	—	TV1A	XAMI 3	BML	—	TABP 11	A 11	LA 11	LXY 011	LXY 1,11	LXY 2,11	LXY 3,11	BM	B
1100	C	TYA	CMA	—	BL	RB 0	SB 0	XAMD 0	BML	—	TABP 12	A 12	LA 12	LXY 0,12	LXY 1,12	LXY 2,12	LXY 3,12	BM	B
1101	D	POF	RAR	—	BL	RB 1	SB 1	XAMD 1	BML	—	TABP 13	A 13	LA 13	LXY 0,13	LXY 1,13	LXY 2,13	LXY 3,13	BM	B
1110	E	TBA	TAB	—	BL	RB 2	SB 2	XAMD 2	BML	TPU1A	TABP 14	A 14	LA 14	LXY 0,14	LXY 1,14	LXY 2,14	LXY 3,14	BM	B
1111	F	WRST	TAY	SZC	BL	RB 3	SB 3	XAMD 3	BML	TPU0A	TABP 15	A 15	LA 15	LXY 0,15	LXY 1,15	LXY 2,15	LXY 3,15	BM	B

The above table shows the relationship between machine language codes and machine language instructions. D_3–D_0 show the low-order 4 bits of the machine language code, and D_8–D_4 show the high-order 5 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use the code marked "–."

The codes for the second word of a two-word instruction are described below.

	The second word
BL	1 1 a a a a a a a a
BML	1 0 a a a a a a a a
BA	1 1 a a a a a a a a
BLA	1 1 a a a p p p p
BMLA	1 0 a a a p p p p
SEA	0 1 0 1 1 n n n n
SZD	0 0 0 1 0 1 0 1 1

REGISTER STRUCTURE

Timer control register V1		at reset : 000_2	at RAM back-up : 000_2	W
$V1_2$	Carrier wave output auto-control bit	0	Auto-control output by timer 1 is invalid	
		1	Auto-control output by timer 1 is valid	
$V1_1$	Timer 1 count source selection bit	0	Carrier wave output (CARRY)	
		1	Bit 5 of watchdog timer (WDT)	
$V1_0$	Timer 1 control bit	0	Stop (Timer 1 state retained)	
		1	Operating	

Timer control register V2		at reset : 0000_2	at RAM back-up : 0000_2	W
$V2_3$	Carrier wave “H” interval expansion bit	0	To expand “H” interval is invalid	
		1	To expand “H” interval is valid (when $V2_2$=1 selected)	
$V2_2$	Carrier wave generation function control bit	0	Carrier wave generation function invalid	
		1	Carrier wave generation function valid	
$V2_1$	Timer 2 count source selection bit	0	f(X_{IN})	
		1	f(X_{IN})/2	
$V2_0$	Timer 2 control bit	0	Stop (Timer 2 state retained)	
		1	Operating	

Logic operation selection register LO		at reset : 00_2	at RAM back-up : 00_2	W
LO_1	Logic operation selection bits	LO_1	LO_0	Logic operation function
LO_0		0	0	Exclusive logic OR operation (XOR)
		0	1	OR operation (OR)
		1	0	AND operation (AND)
		1	1	Not available

Pull-down control register PU0		at reset : 0000_2	at RAM back-up : state retained	W
$PU0_3$	Ports G_2, G_3 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU0_2$	Ports G_0, G_1 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU0_1$	Port E_1 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU0_0$	Port E_0 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	

Pull-down control register PU1		at reset : 0000_2	at RAM back-up : state retained	W
$PU1_3$	Port D_7 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU1_2$	Port D_6 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU1_1$	Port D_5 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU1_0$	Port D_4 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{DD}	Supply voltage		–0.3 to 5	V
V_I	Input voltage		–0.3 to V_{DD}+0.3	V
V_O	Output voltage		–0.3 to V_{DD}+0.3	V
P_d	Power dissipation	Ta = 25 °C	300	mW
T_{opr}	Operating temperature range		–20 to 85	°C
T_{stg}	Storage temperature range		–40 to 125	°C

RECOMMENDED OPERATING CONDITIONS

(Ta = –20 °C to 85 °C, V_{DD} = 1.8 V to 3.6 V, unless otherwise noted)

Symbol	Parameter		Conditions	Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage			1.8		3.6	V
V_{RAM}	RAM back-up voltage (at RAM back-up mode)			1.1		3.6	V
V_{SS}	Supply voltage				0		V
V_{IH}	“H” level input voltage Ports D_4–D_7, E, G		V_{DD} = 3.0 V	0.7V_{DD}		V_{DD}	V
V_{IH}	“H” level input voltage X_{IN}		V_{DD} = 3.0 V	0.8V_{DD}		V_{DD}	V
V_{IL}	“L” level input voltage Ports D_4–D_7, E, G		V_{DD} = 3.0 V	0		0.2V_{DD}	V
V_{IL}	“L” level input voltage X_{IN}		V_{DD} = 3.0 V	0		0.2V_{DD}	V
I_{OH}(peak)	“H” level peak output current Ports D, E_1, G		V_{DD} = 3.0 V			–4	mA
I_{OH}(peak)	“H” level peak output current Port E_0		V_{DD} = 3.0 V			–24	mA
I_{OH}(peak)	“H” level peak output current CARR		V_{DD} = 3.0 V			–20	mA
I_{OL}(peak)	“L” level peak output current CARR		V_{DD} = 3.0 V			4	mA
I_{OH}(avg)	“H” level average output current Ports D, E_1, G		V_{DD} = 3.0 V			–2	mA
I_{OH}(avg)	“H” level average output current Port E_0		V_{DD} = 3.0 V			–12	mA
I_{OH}(avg)	“H” level average output current CARR		V_{DD} = 3.0 V			–10	mA
I_{OL}(avg)	“L” level average output current CARR		V_{DD} = 3.0 V			2	mA
f(X_{IN})	System clock frequency	when STCK = f(X_{IN})/8 selected	Ceramic resonance			4	MHz
		when STCK = f(X_{IN}) selected	Ceramic resonance			500	kHz
V_{DET}	Voltage drop detection circuit detection voltage			1.10		1.80	V
			Ta=25 °C	1.40	1.50	1.56	
T_{DET}	Voltage drop detection circuit low voltage determination time		When supply voltage passes the detected voltage at ±50V/s.		0.2	1.2	ms
T_{PON}	Power-on reset circuit valid power source rising time		V_{DD} = 0 to 2.2 V			1	ms

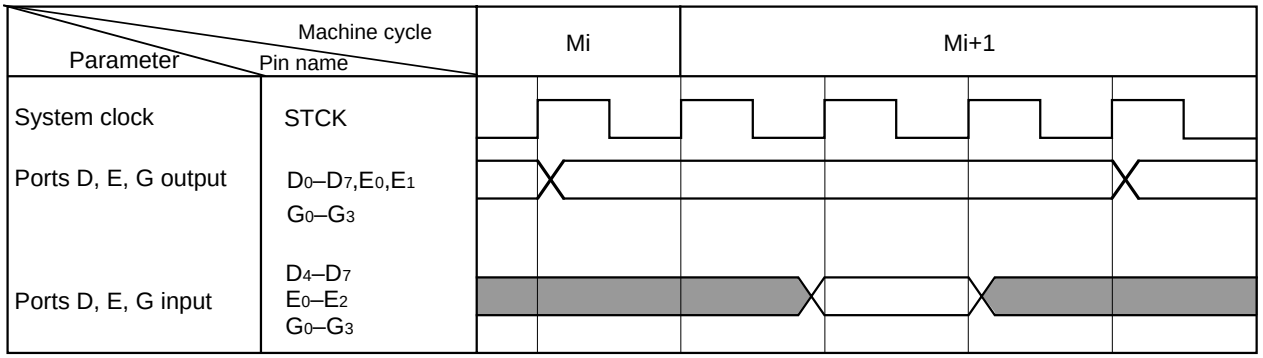
Note: The average output current ratings are the average current value during 100 ms.

ELECTRICAL CHARACTERISTICS

(Ta = –20 °C to 85 °C, V_{DD} = 3 V, unless otherwise noted)

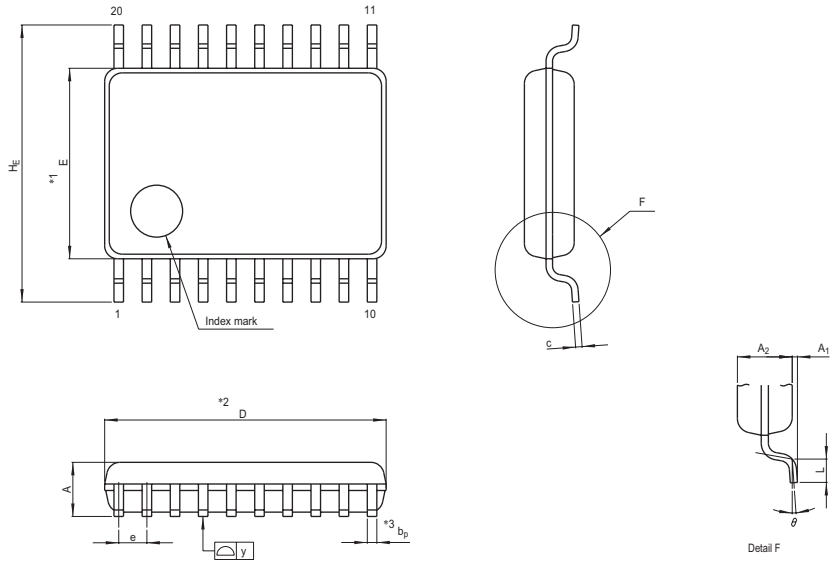
Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OL}	"L" level output voltage Port CARR	I_{OL} = 2 mA			0.9	V
V_{OL}	"L" level output voltage X_{OUT}	I_{OL} = 0.2 mA			0.9	V
V_{OH}	"H" level output voltage Ports D, E_1, G	I_{OH} = –2 mA	2.1			V
V_{OH}	"H" level output voltage Port E_0	I_{OH} = –12 mA	1.5			V
V_{OH}	"H" level output voltage CARR	I_{OH} = –10 mA	1.0			V
V_{OH}	"H" level output voltage X_{OUT}	I_{OH} = –0.2 mA	2.1			V
I_{IL}	"L" level input current Ports D_4–D_7, E, G	V_I = V_{SS}			–1	μA
I_{IH}	"H" level input current Ports E_0, E_1	V_I = V_{DD} Pull-down transistor in off-state			1	μA
I_{OZ}	Output current at off-state Ports D, E_0, E_1, G	V_O = V_{SS}			–1	μA
I_{DD}	Supply current (when operating)	f(X_{IN}) = 4.0 MHz		400	800	μA
		f(X_{IN}) = 500 kHz		250	500	μA
	Supply current (at RAM back-up)			1	3	μA
		Ta = 25 °C		0.1	0.5	μA
R_{PH}	Pull-down resistor value Ports D_4–D_7, E, G	V_{DD} = 3 V, V_I = 3 V	75	150	300	kΩ
R_{OSC}	Feedback resistor value between X_{IN}–X_{OUT}		700		3200	kΩ

BASIC TIMING DIAGRAM



PACKAGE OUTLINE

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LSSOP20-4.4x6.5-0.65	PLSP0020JB-A	20P2F-A	0.1g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.4	6.5	6.6
E	4.3	4.4	4.5
A_2	—	1.15	—
A	—	—	1.45
A_1	0	0.1	0.2
b_p	0.17	0.22	0.32
c	0.13	0.15	0.2
θ	0°	—	10°
H_E	6.2	6.4	6.6
e	0.53	0.65	0.77
y	—	—	0.10
L	0.3	0.5	0.7

REVISION HISTORY	4283 Group Data Sheet

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 07, 2005	–	First edition issued.
1.01	Mar. 20, 2006	24	The followings of LIST OF PRECAUTIONS revised. (12)Overvoltage → (12)QzROM revised. (13)Notes On ROM Code Protect added.
		→	Pages 27, 38, 52-55: SNZT1 and SNZT2 revised.
		62	Package outline revised.

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

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.

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

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

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

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

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

© 2006. Renesas Technology Corp., All rights reserved. Printed in Japan.

Colophon .6.0