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.

R8C/35A Group

I²C Version Program Downloader

1. Abstract

This documents describes the I²C version program downloader for the R8C/35A Group.

2. Introduction

The application example described in this document applies to the following MCU and parameter:

- MCU: R8C/35A Group
- XIN clock frequency: 20 MHz

The sample program may include operations of unused bit functions for the convenience of the SFR bit layout. Set the values according to the operating conditions of the user system.

3. Program Downloader Overview

3.1 Downloader Specifications

- The system program (including program downloader process) is allocated to block 0.
- The program downloader erases and writes mainly to user programs other than the user program in block 0. The program downloader ignores rewrite operations to block 0.
- EW0 mode is used by the program downloader for rewriting the CPU.
- In a reset start, the program downloader checks the state of port P3_3 and selects either to use the program downloader or the user program. The program downloader operates when port P3_3 is high, and the user program operates when port P3_3 is low.
- The virtual fixed vector table is allocated to block 1 to use the fixed vector table interrupt in the user program.
- I^2C-bus slave mode is used to communicate with a programmer.
- 1010101b is used for the slave address.
- Refer to 4. Downloader Communication Protocol for the communication protocol.

Figure 3.1 shows an example of a Connection, Figure 3.2 shows the Transfer Format, Figure 3.3 shows the Memory Map (32 Kbyte ROM MCU), and Figure 3.4 shows an example of the System Interrupt Operation (Overflow Interrupt).

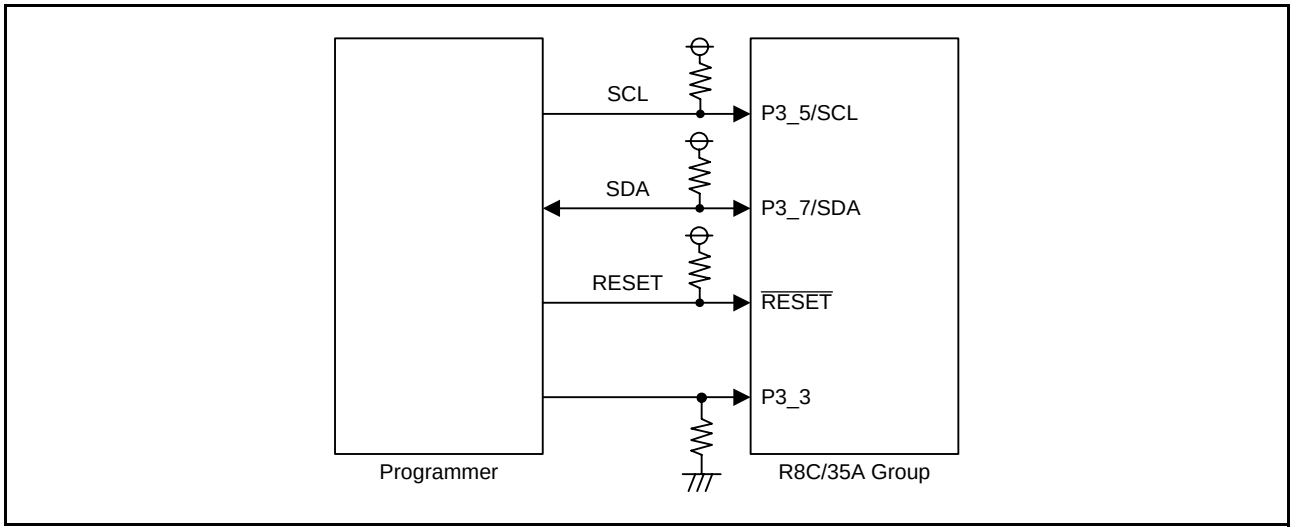


Figure 3.1 Connection

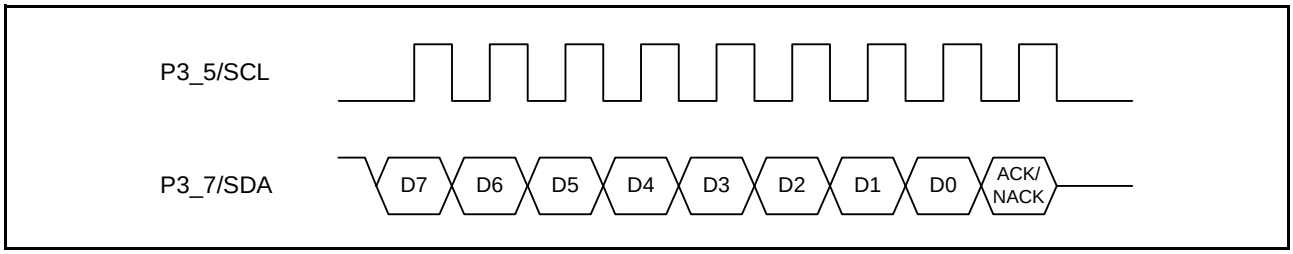
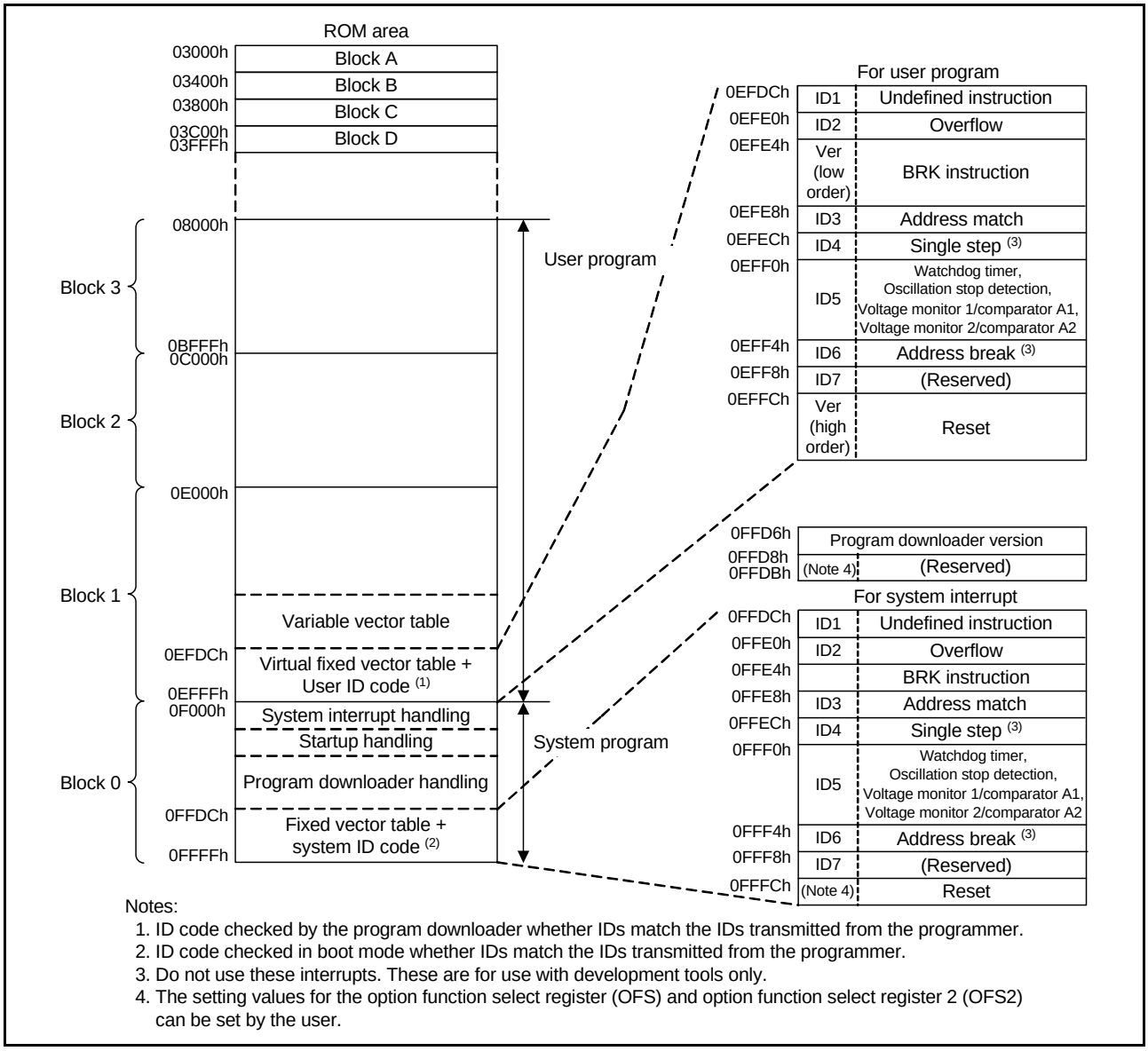


Figure 3.2 Transfer Format



Notes:
1. ID code checked by the program downloader whether IDs match the IDs transmitted from the programmer.
2. ID code checked in boot mode whether IDs match the IDs transmitted from the programmer.
3. Do not use these interrupts. These are for use with development tools only.
4. The setting values for the option function select register (OFS) and option function select register 2 (OFS2) can be set by the user.

Figure 3.3 Memory Map (32 Kbyte ROM MCU)

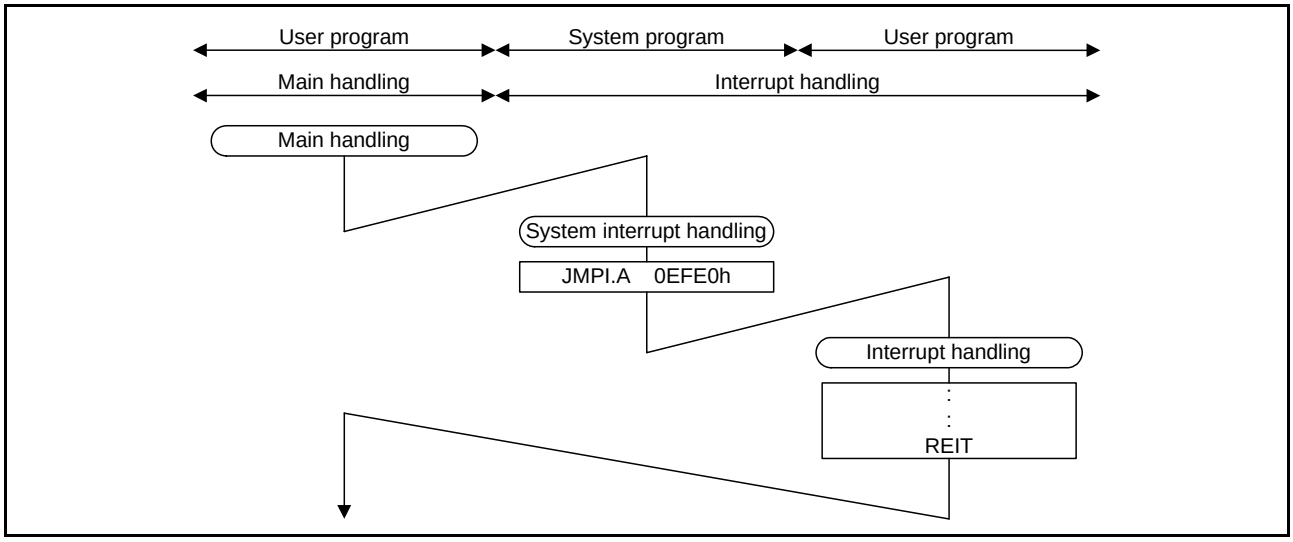
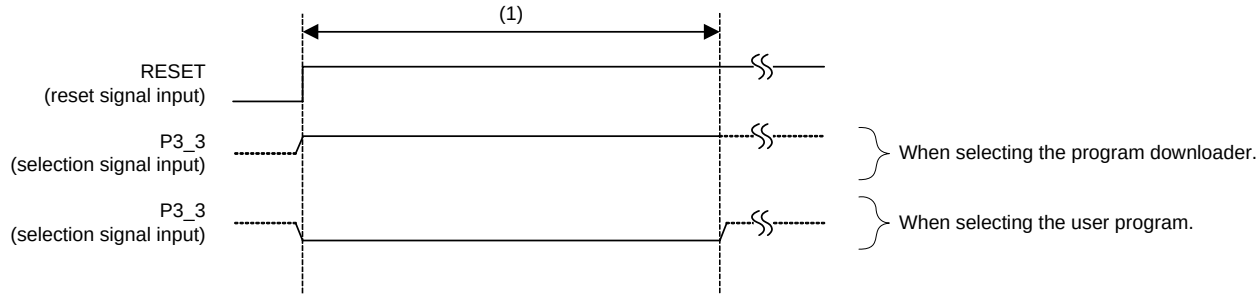


Figure 3.4 System Interrupt Operation (Overflow Interrupt)

3.2 Timing after Reset

When the program downloader is activated, input a high to P3_3 pin before reset is deasserted and retain high until the program downloader is selected.



(1) While the program downloader or user program selection signal is held: Approximately 3.5 ms. (1)
The reset sequence is included in above time.

Note:
1. Time when the low-speed on-chip oscillator (60 kHz (min)) operates.

Figure 3.5 Signal Control Timing after Reset

3.3 Initial Settings

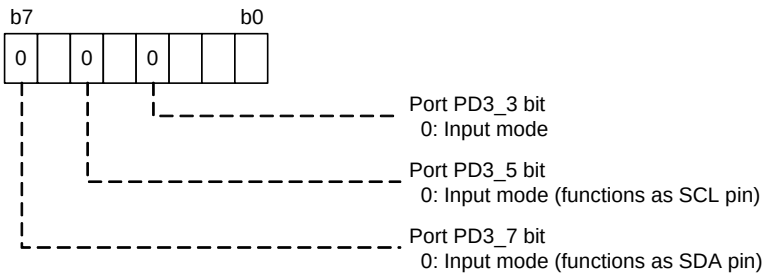
(1) Option function select register (OFS)
The OFS register is assigned to the highest-order address 0FFFFh in the fixed vector table. Set the OFS register by a program of the program downloader.

(2) Option function select register 2 (OFS2)
The OFS2 register is assigned to 0FFDBh in the reserved area. Set the OFS2 register by a program of the program downloader.

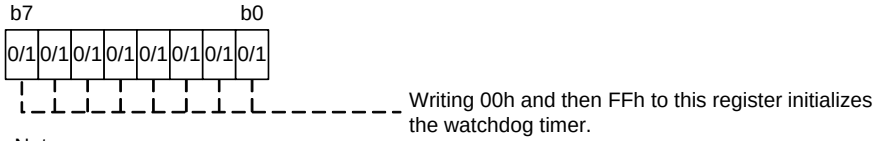
(3) Watchdog timer
When using the watchdog timer, enable the WDT_USE definition in the fla_r835a.inc file.

3.4 Registers

Port P3 Direction Register PD3 [address 00E7h]

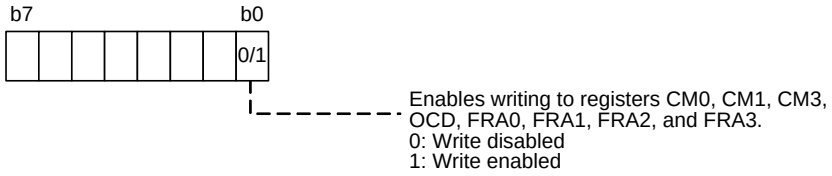


Watchdog Timer Reset Register WDTR [address 000Dh]

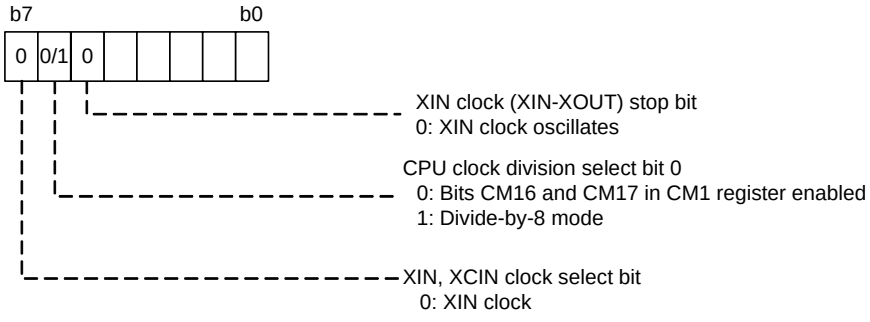


Note:
1. This setting is unnecessary when the watchdog timer is not used.

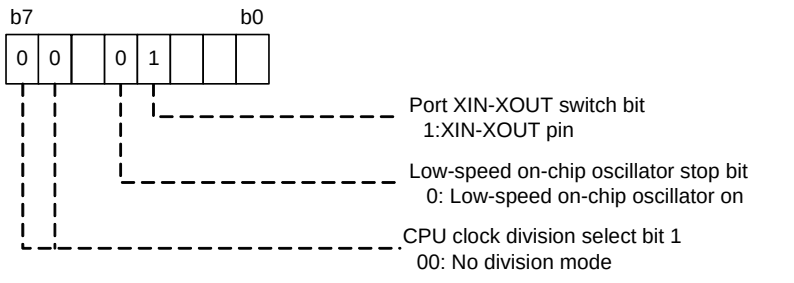
Protect Register PRCR [address 000Ah]



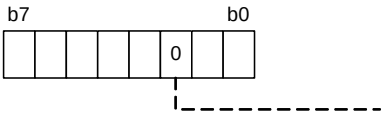
System Clock Control Register 0 CM0 [address 0006h]



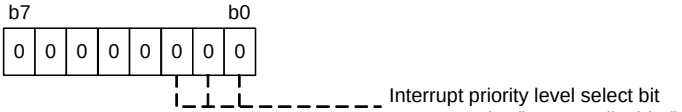
System Clock Control Register 1 CM1 [address 0007h]



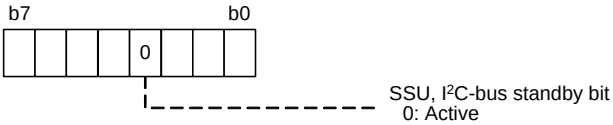
Oscillation Stop Detection Register OCD [address 000Ch]



Interrupt Control Register IICIC [address 004Fh]

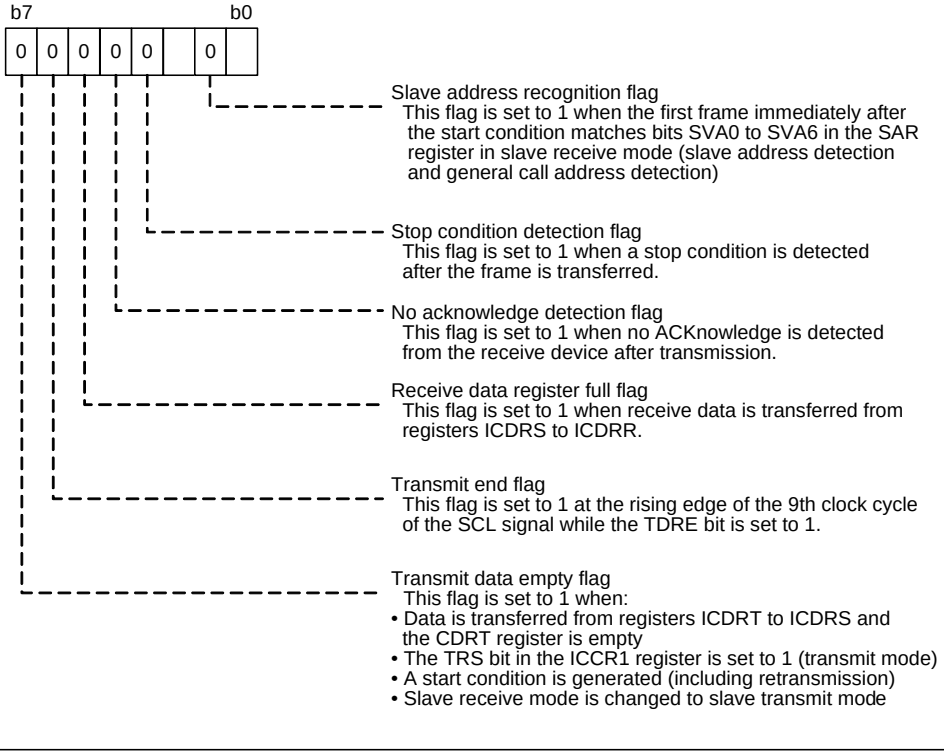


Module Standby Control Register MSTCR [address 0008h]

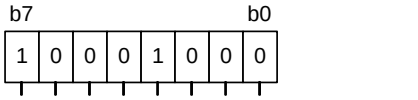


SSU/IIC Pin Select Register SSUIICSR [address 018Ch]

IIC bus Status Register ICSR [address 019Ch]



IIC bus Control Register 1 ICCR1 [address 0198h]



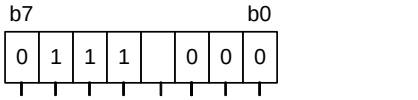
Transmit clock select bits 3 to 0
This bit is used for maintaining the setup time in transmit mode of slave mode. The time is 10Tcyc when the CKS3 bit is set to 0 and 20Tcyc when the CKS3 bit is set to 1. (1Tcyc = 1/f1(s))

Transfer/receive select bit, Master/slave select bit
00: Slave Receive Mode

Receive disable bit
After reading the ICDRR register while the TRS bit is set to 0
0: Next receive operation continues

I2C-bus interface enable bit
1: This module is enabled for transfer operations
(Pins SCL and SDA are in a bus drive state)

IIC bus Control Register 2 ICCR2 [address 0199h]

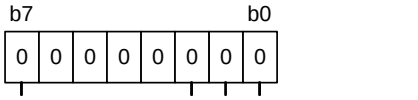


Set each bit to initialize.

SCL monitor flag
0: SCL pin is set to “L”.
1: SCL pin is set to “H”.

Set each bit to initialize.

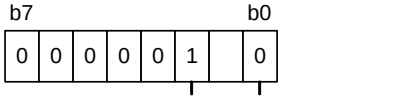
IIC bus Mode Register ICMR [address 019Ah]



Bit counters 2 to 0
000: 9 bits

MSB-first/LSB-first select bit
0: Data transfer with MSB-first

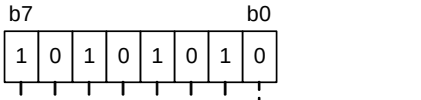
IIC Interrupt Enable Register ICIER [address 019Bh]



Transmit acknowledge select bit
0: In receive mode, 0 is transmitted as the acknowledge bit.

Acknowledge bit detection select bit
1: When the receive acknowledge bit is set to 1, continuous transfer is halted.

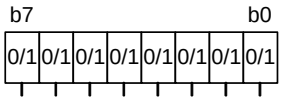
Slave Address Register SAR [address 019Dh]



Format select bit
0: I²C-bus format

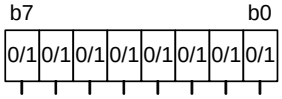
Set an address different from that of the other slave devices connected to the I²C-bus.
When the 7 high-order bits of the first frame transmitted after the start condition match bits SVA0 to SVA6 in slave mode of the I²C-bus format, the MCU operates as a slave device.

IIC bus Transmit Data Register 1 ICDRT [address 0194h]



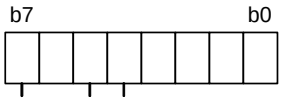
This register stores transmit data.
When the ICDRS register is detected as empty, the stored transmit data item is transferred to the ICDRS register and data transmission starts. When the next unit of transmit data is written to the ICDRT register while data is transmitted to the ICDRS register, continuous transmission is enabled.

IIC bus Receive Data Register 1 ICDRR [address 0196h]



This register stores receive data.
When the ICDRS register receives 1 byte of data, the receive data is transferred to the ICDRR register and the next receive operation is enabled.

Flash Memory Status Register FST [address 01B2h]



Program error flag
0: No program error
1: Program error

Erase error/blank check error flag
0: No erase error/blank check error
1: Erase error/blank check error

Ready/busy status flag
0: Busy
1: Ready

Flash Memory Control Register 0 FMR0 [address 01B4h]

b7							b0
0	0	0	0	0	0	0/1	0

CPU rewrite mode select bit
0: CPU rewrite mode disabled
1: CPU rewrite mode enabled

EW1 mode select bit
0: EW0 mode

Flash memory stop bit
0: Flash memory operates.

Erase/write error interrupt enable bit
0: Erase/write error interrupt disabled

Flash access error interrupt enable bit
0: Flash access error interrupt disabled

Flash ready status interrupt enable bit
0: Flash ready status interrupt disabled

Flash Memory Control Register 1 FMR1 [address 01B5h]

b7							b0
0/1	0/1	0/1	0/1	0/1	0	0	0

Lock bit disable select bit
0: Lock bit enabled
1: Lock bit disabled

Data flash block A rewrite disable bit
0: Rewrite enabled (software command acceptable)
1: Rewrite disabled (software command not acceptable, no error occurred)

Data flash block B rewrite disable bit
0: Rewrite enabled (software command acceptable)
1: Rewrite disabled (software command not acceptable, no error occurred)

Data flash block C rewrite disable bit
0: Rewrite enabled (software command acceptable)
1: Rewrite disabled (software command not acceptable, no error occurred)

Data flash block D rewrite disable bit
0: Rewrite enabled (software command acceptable)
1: Rewrite disabled (software command not acceptable, no error occurred)

3.5 Memory

Table 3.1 Memory

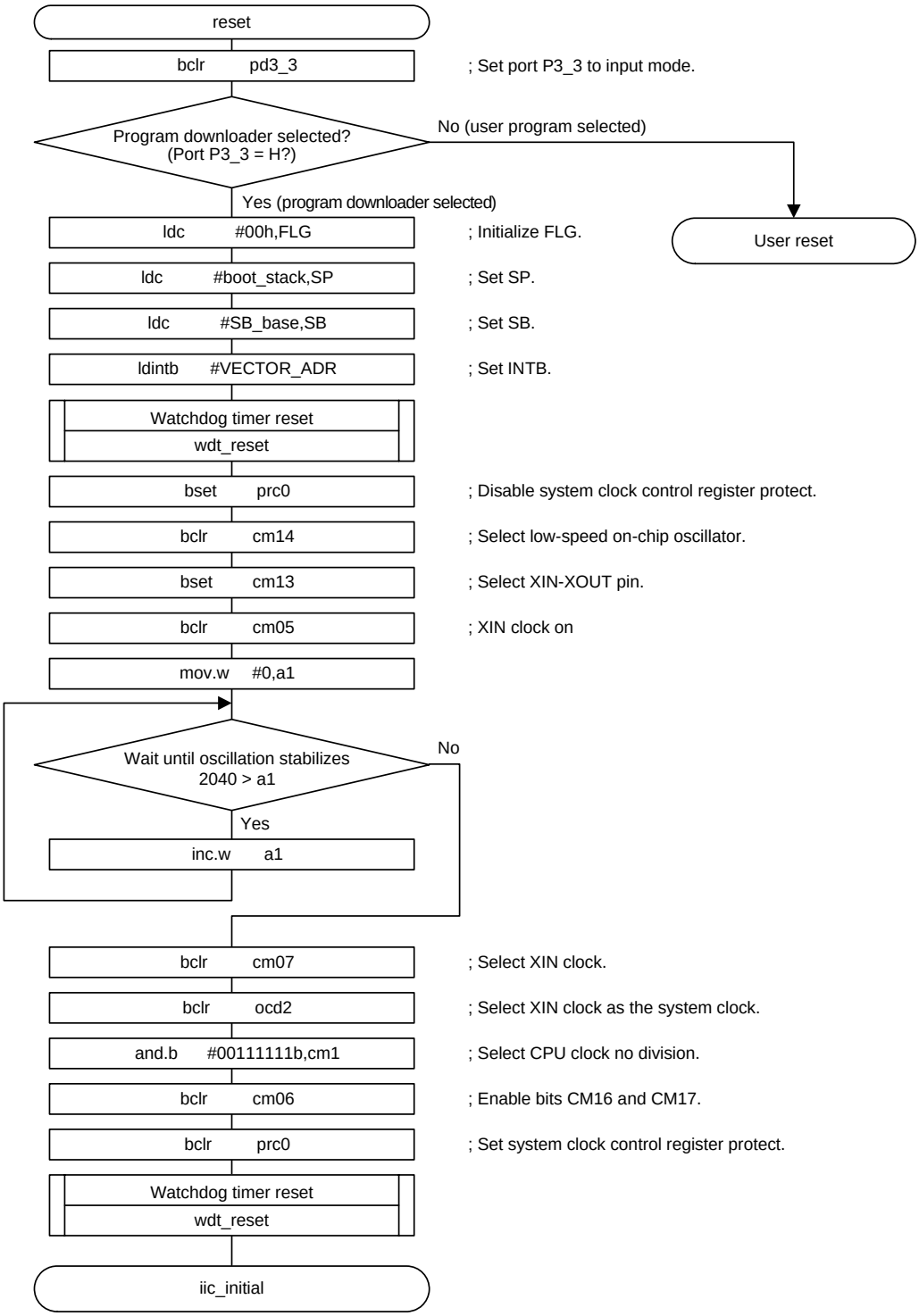
Assigned Memory	Size	Remarks
ROM	1461 bytes	System program only (including fixed vector and variable vector table)
RAM	427 bytes	System program only

Table 3.2 RAM and Definitions

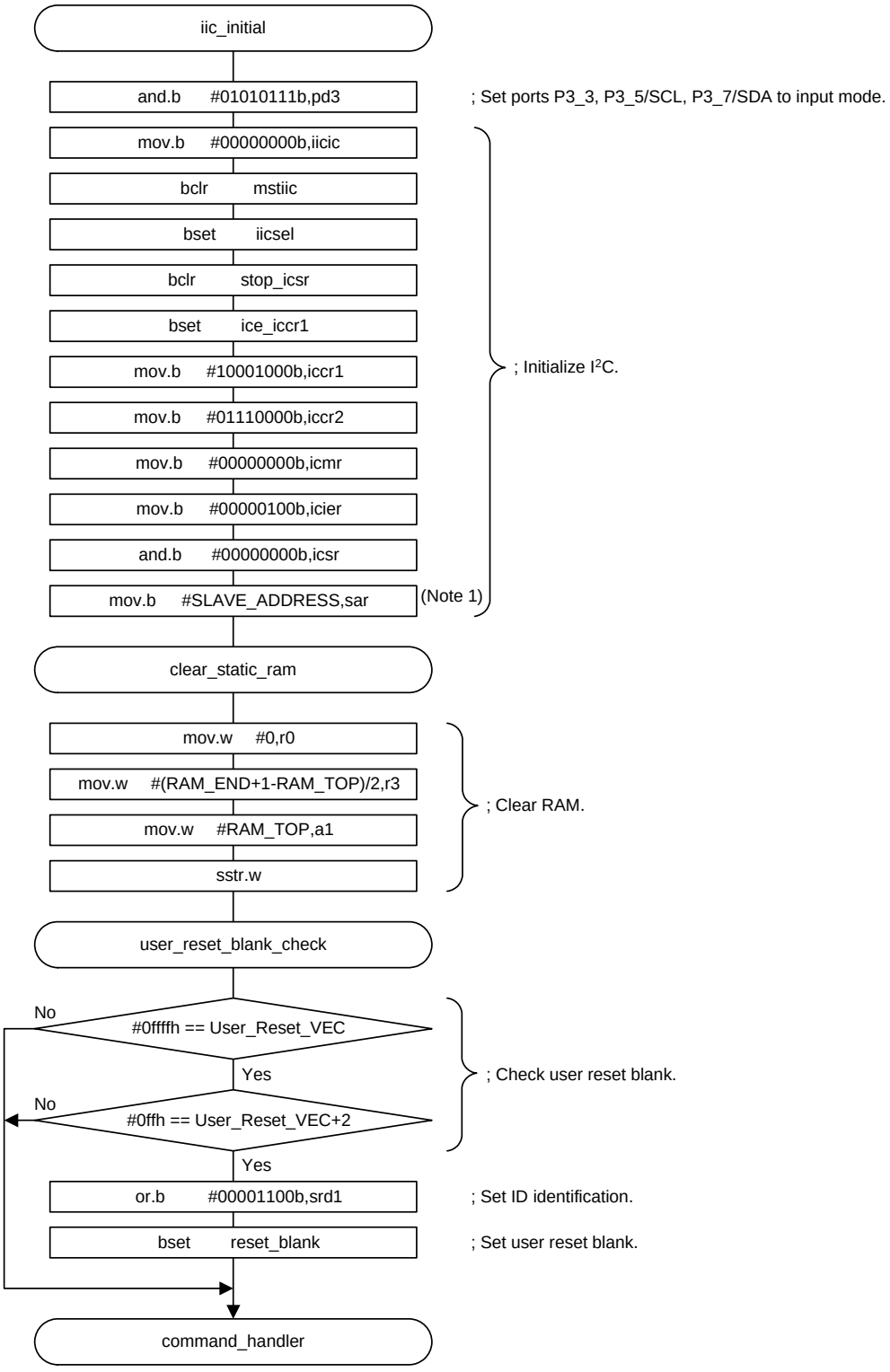
Symbol	Size	Description
ram_execute	128 bytes	EW0 mode program area
status_flags	1 byte	Serial flag area
reset_blank	–	User program blank flag
srd1	1 byte	SRD1 register
srd08	–	SR8 bit
srd09	–	SR9 bit
srd10	–	SR10 bit
srd11	–	SR11 bit
srd12	–	SR12 bit
srd13	–	SR13 bit
srd14	–	SR14 bit
srd15	–	SR15 bit
srd	1 byte	SRD register
address	4 bytes	Address data
temp	32 bytes	Temporary (used with the stack area)
rx_data	1 byte	Receive data
tx_data	1 byte	Transmit data
page_buffer	256 bytes	Page buffer

3.6 Flowchart

(1) Startup handling 1

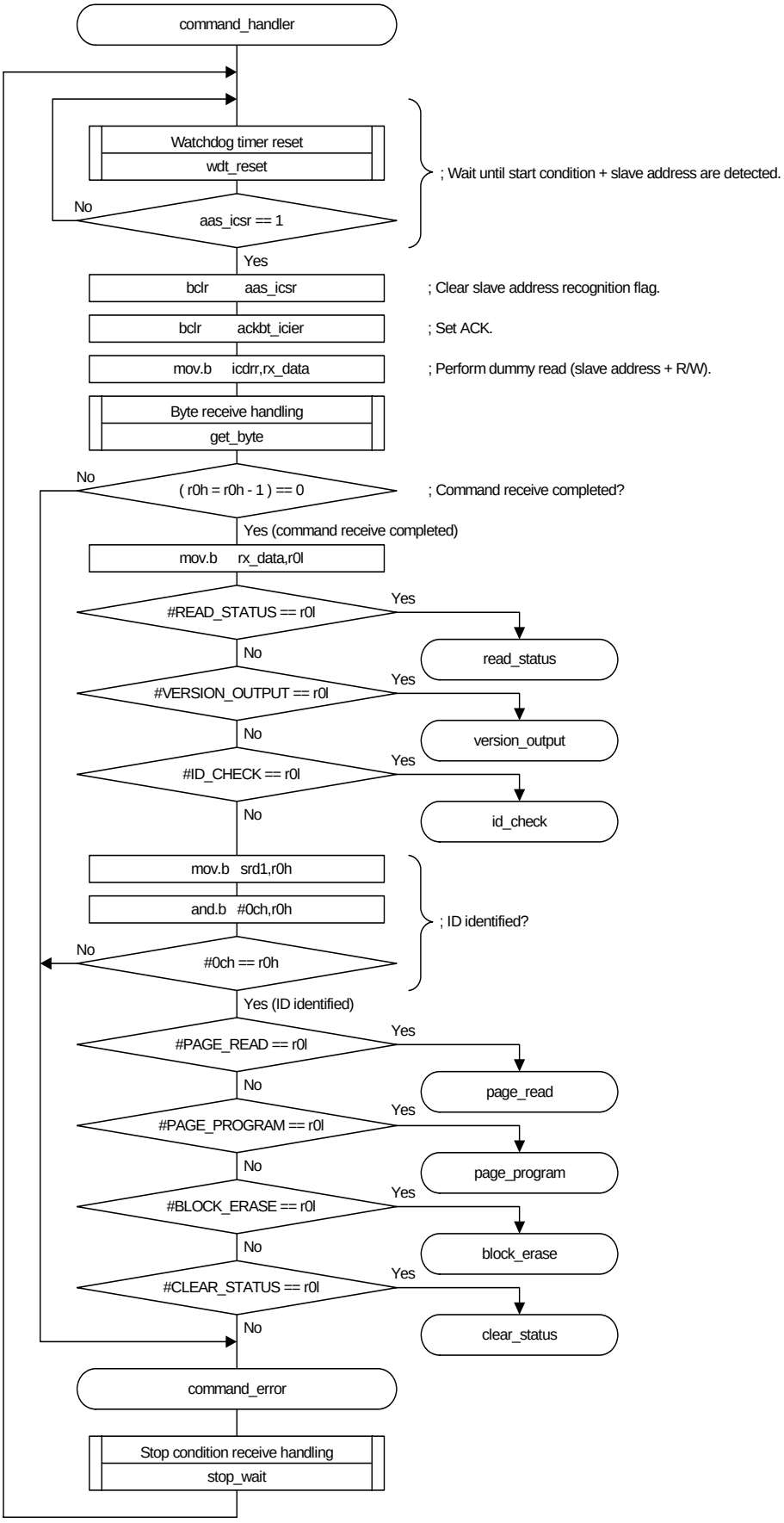


(2) Startup handling 2

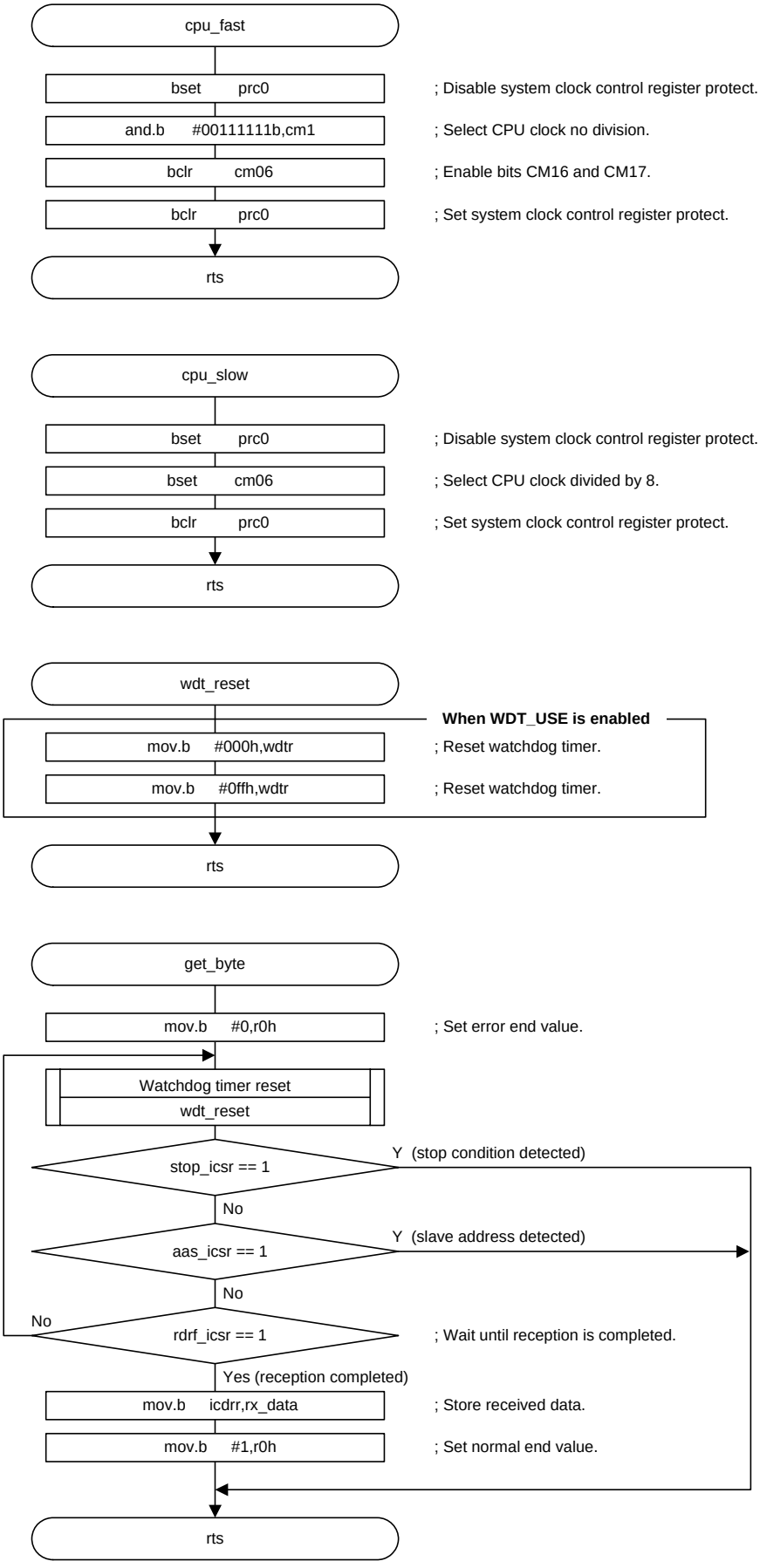


Note:
1. SLAVE_ADDRESS = 0AAh

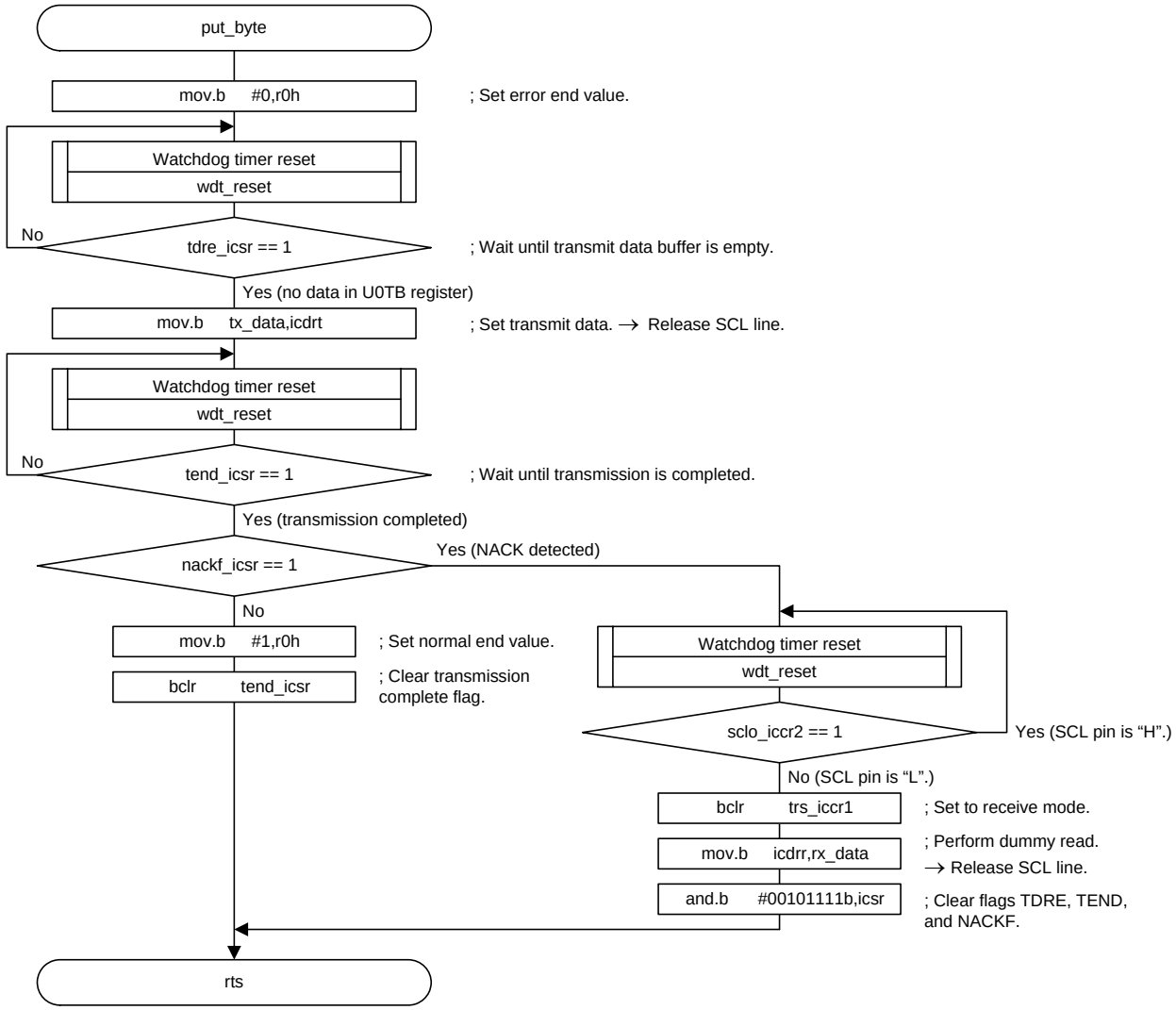
(3) Command handler



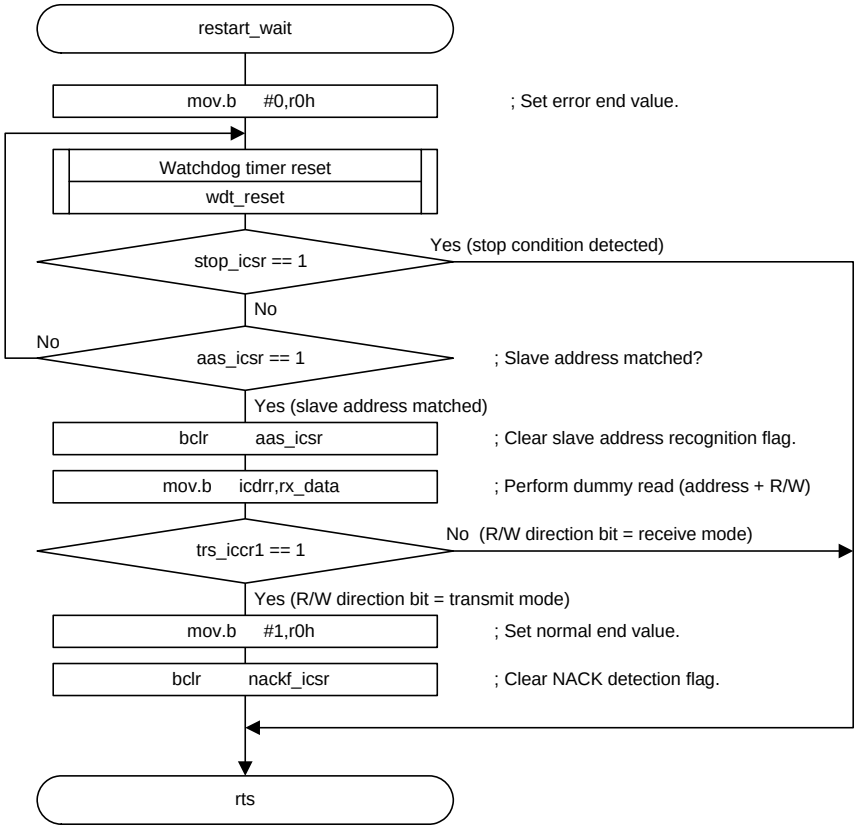
(4) Subroutine 1

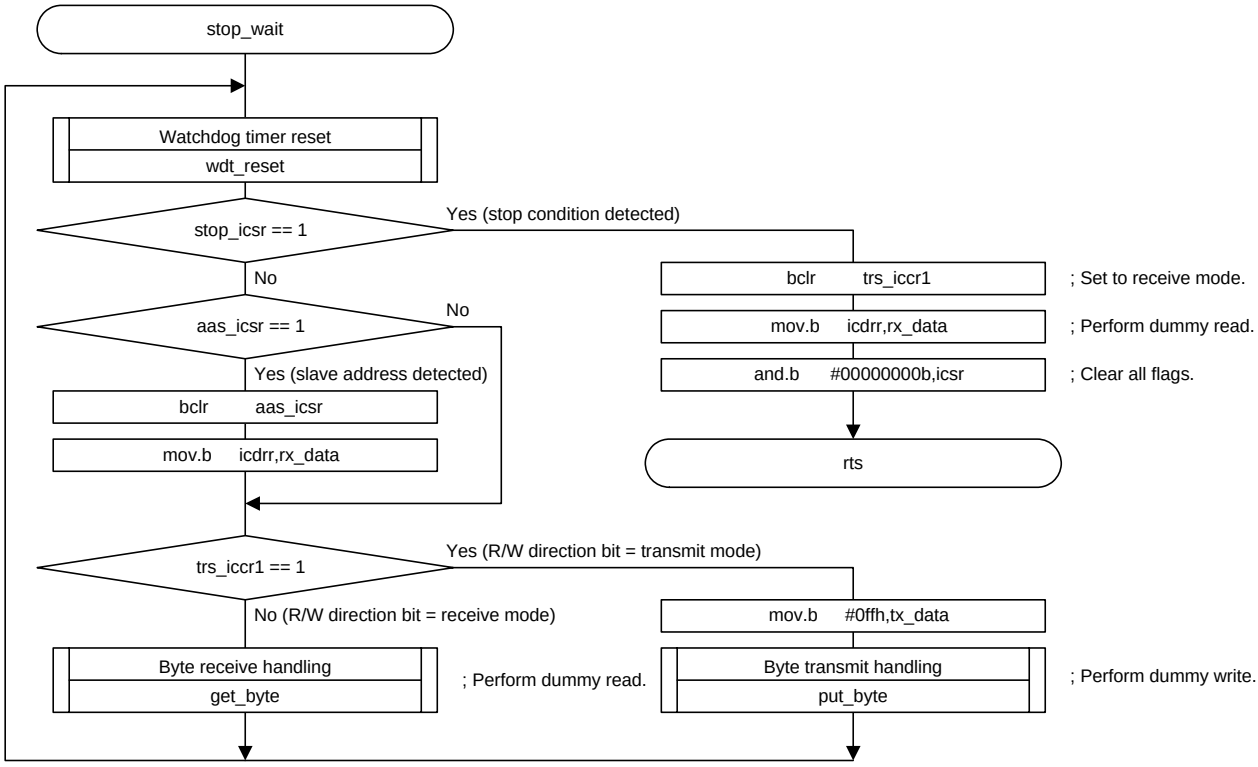


(5) Subroutine 2

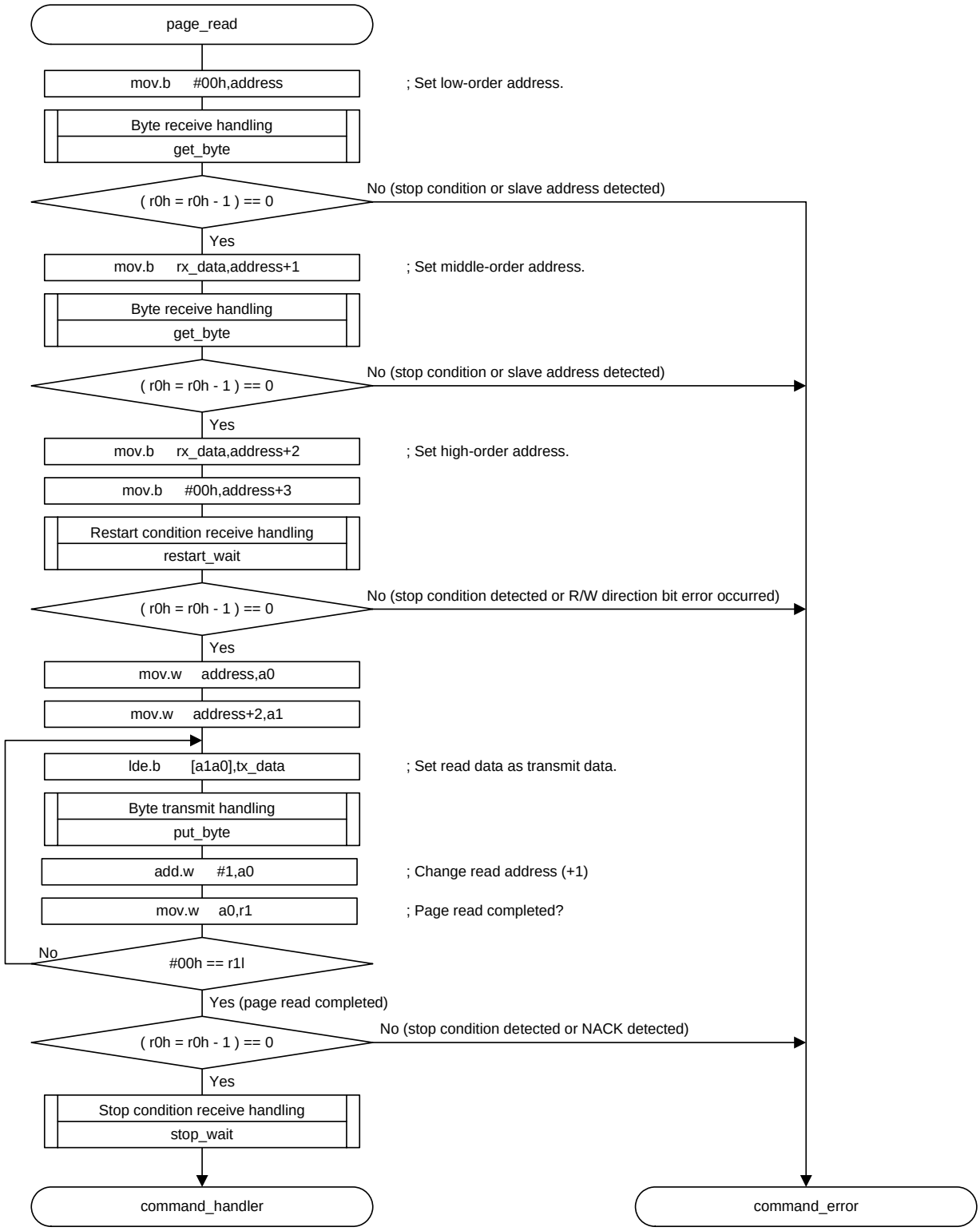


(6) Subroutine 3

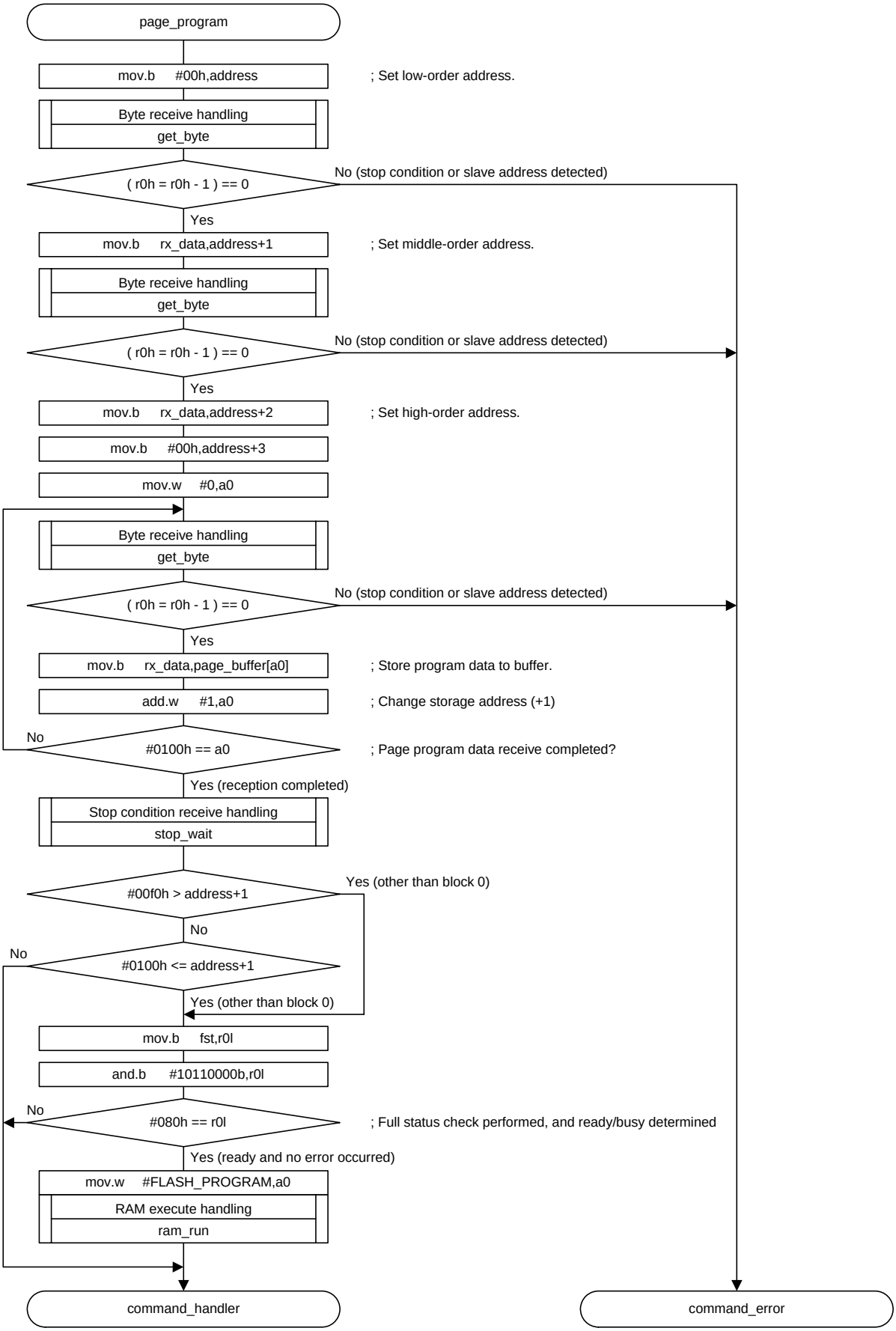




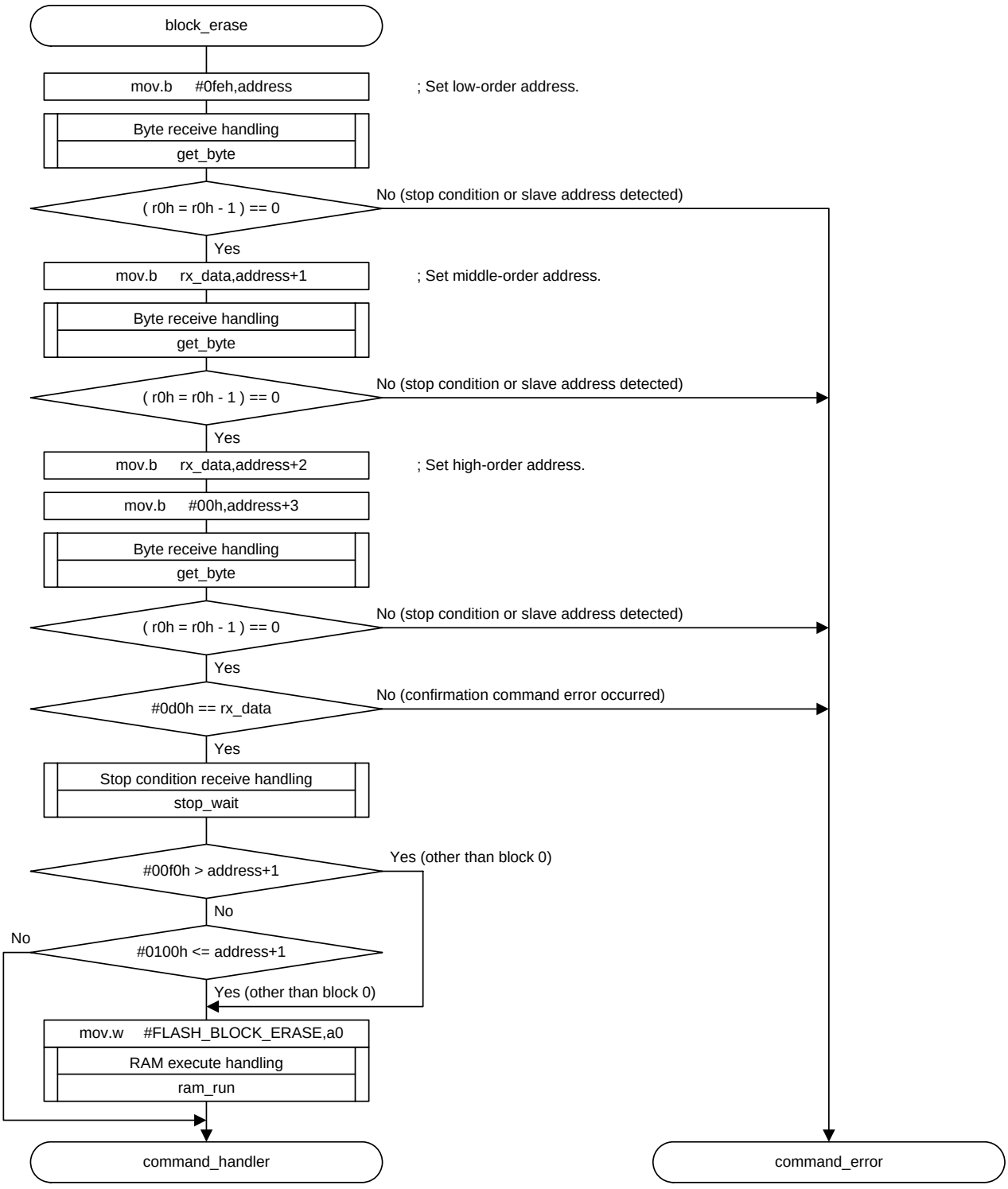
(7) Page read


page_read
mov.b #00h,address
; Set low-order address.
Byte receive handling
get_byte
(r0h = r0h - 1) == 0
No (stop condition or slave address detected)
Yes
mov.b rx_data,address+1
; Set middle-order address.
Byte receive handling
get_byte
(r0h = r0h - 1) == 0
No (stop condition or slave address detected)
Yes
mov.b rx_data,address+2
; Set high-order address.
mov.b #00h,address+3
Restart condition receive handling
restart_wait
(r0h = r0h - 1) == 0
No (stop condition detected or R/W direction bit error occurred)
Yes
mov.w address,a0
mov.w address+2,a1
lde.b [a1a0],tx_data
; Set read data as transmit data.
Byte transmit handling
put_byte
add.w #1,a0
; Change read address (+1)
mov.w a0,r1
; Page read completed?
No
#00h == r1l
Yes (page read completed)
(r0h = r0h - 1) == 0
No (stop condition detected or NACK detected)
Yes
Stop condition receive handling
stop_wait
command_handler
command_error

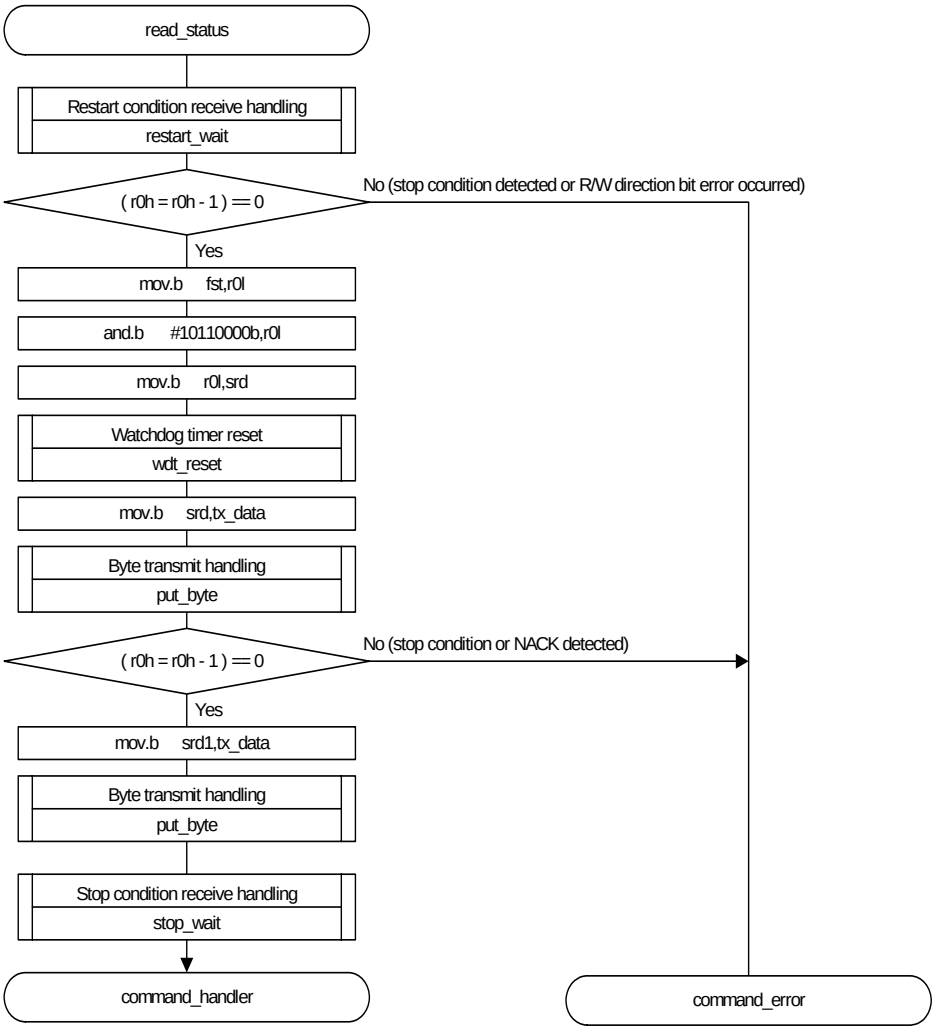
(8) Page program



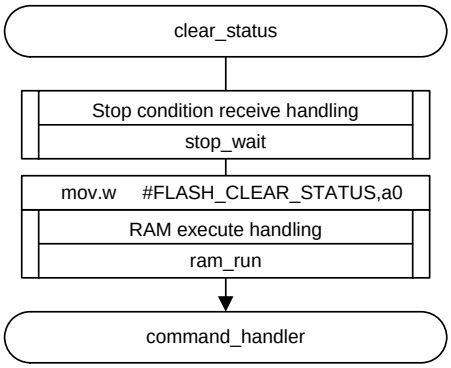
(9) Block erase



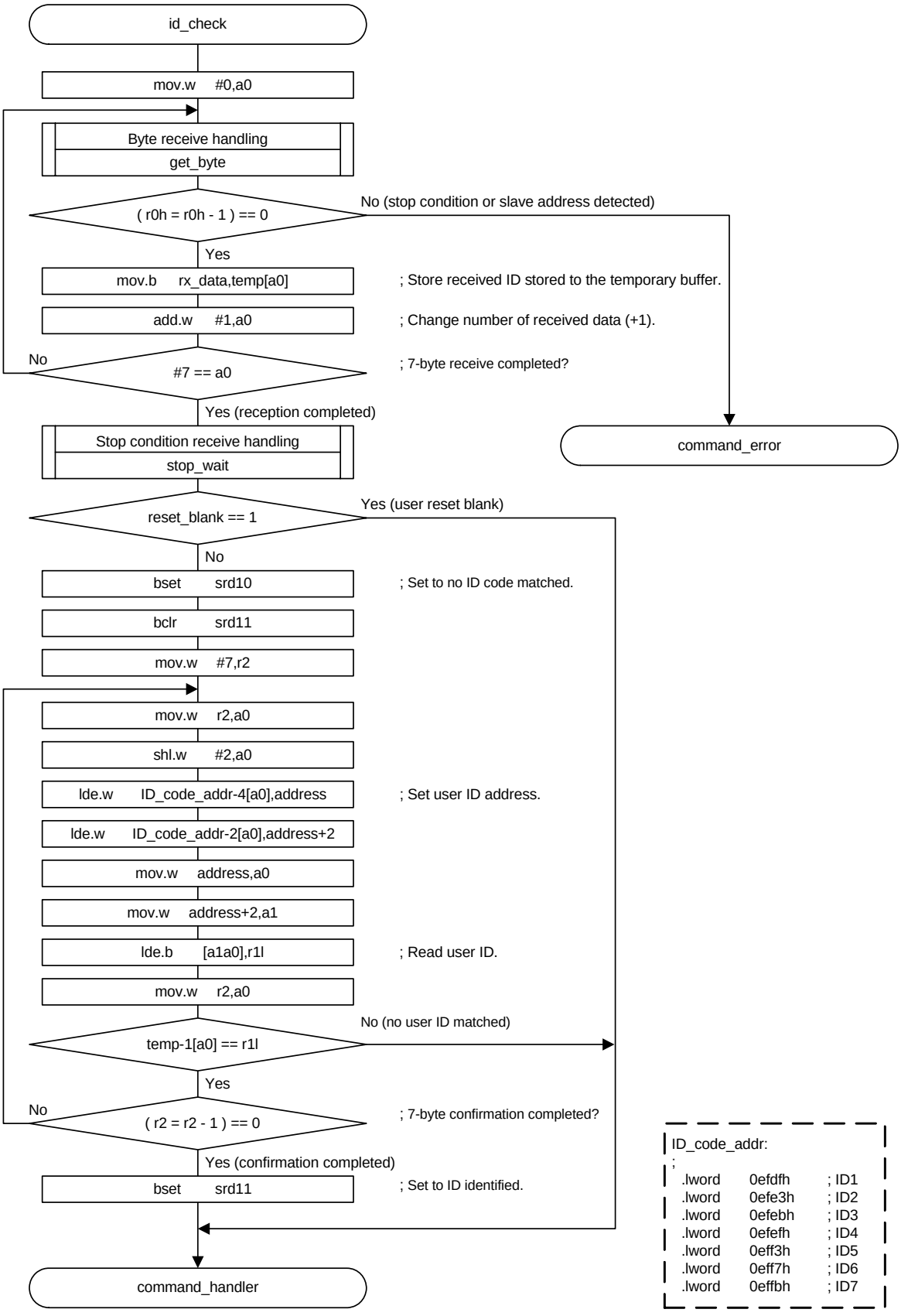
(10) Read status register



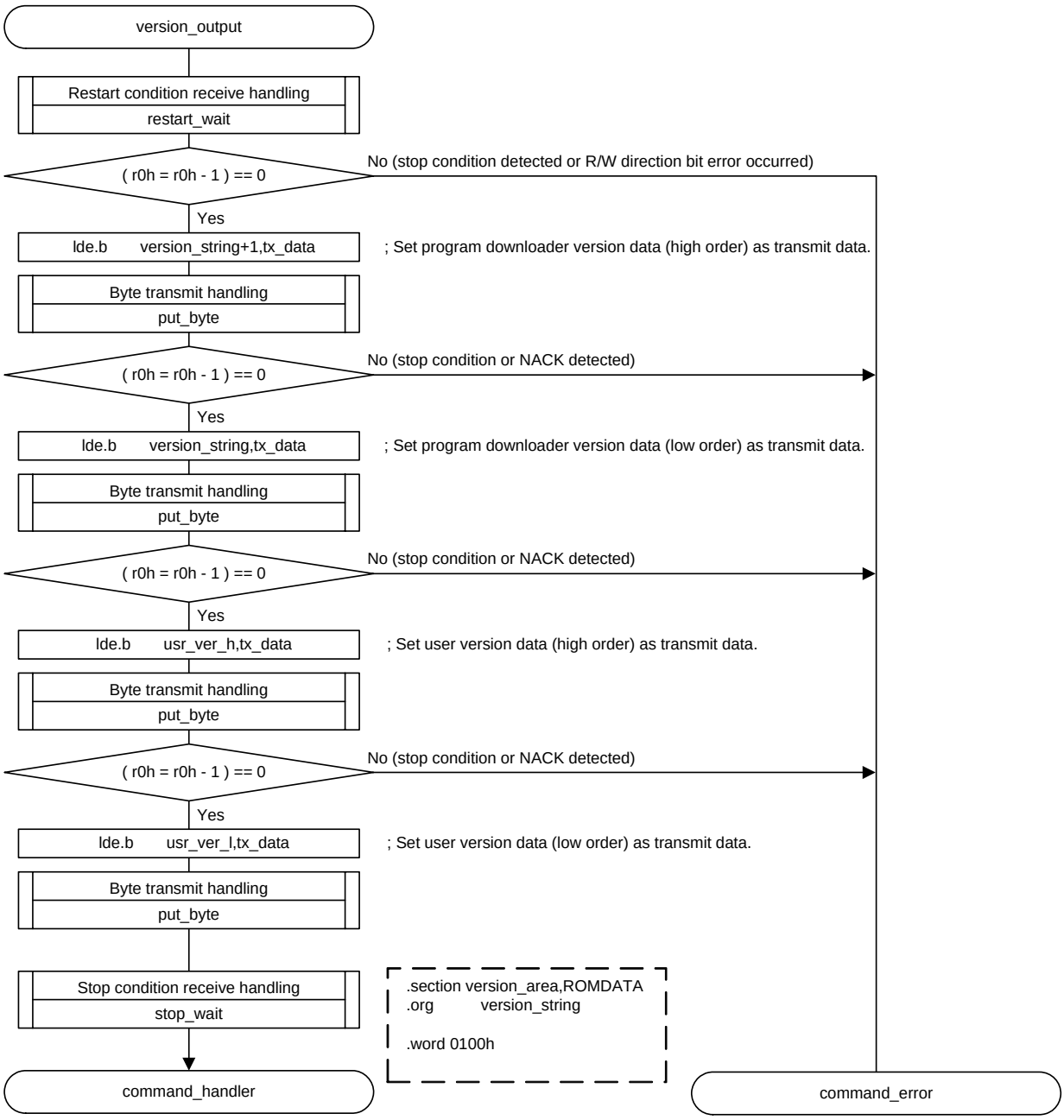
(11) Clear status register



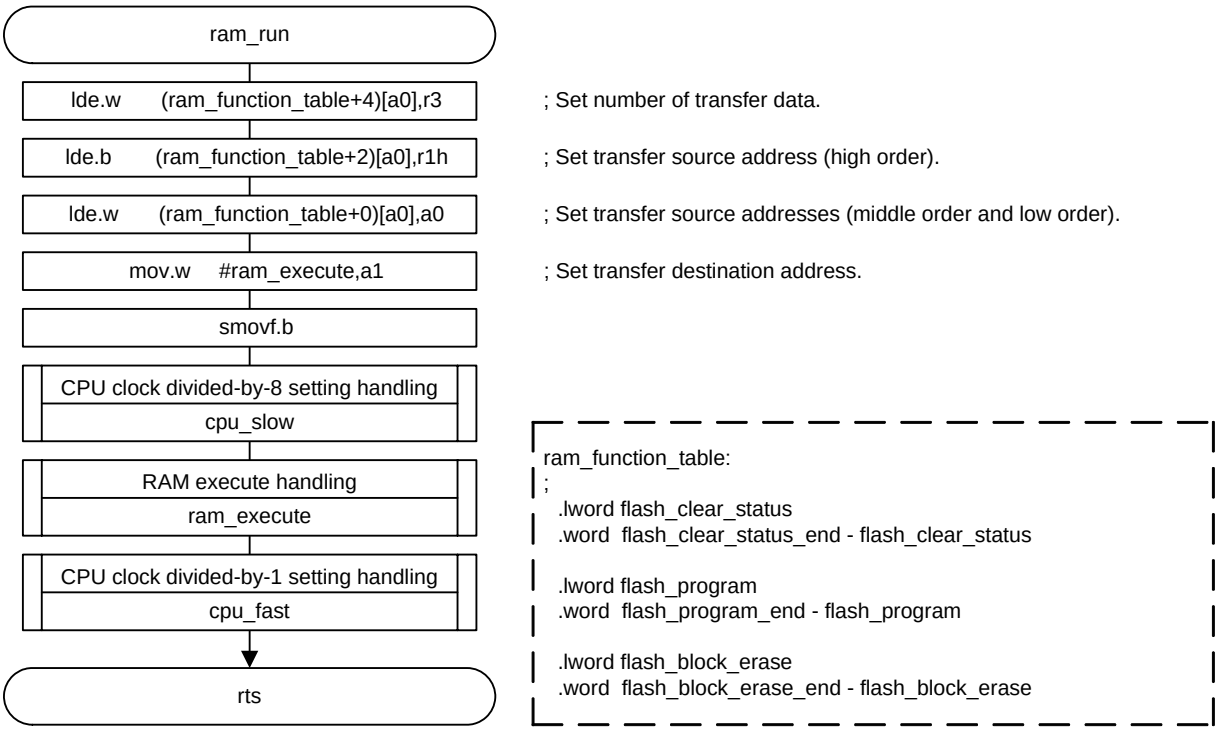
(12) ID check


id_check
mov.w #0,a0
Byte receive handling
get_byte
(r0h = r0h - 1) == 0
No (stop condition or slave address detected)
Yes
mov.b rx_data,temp[a0]
; Store received ID stored to the temporary buffer.
add.w #1,a0
; Change number of received data (+1).
No
#7 == a0
; 7-byte receive completed?
Yes (reception completed)
Stop condition receive handling
stop_wait
command_error
reset_blank == 1
Yes (user reset blank)
No
bset srd10
; Set to no ID code matched.
bclr srd11
mov.w #7,r2
mov.w r2,a0
shl.w #2,a0
lde.w ID_code_addr-4[a0],address
; Set user ID address.
lde.w ID_code_addr-2[a0],address+2
mov.w address,a0
mov.w address+2,a1
lde.b [a1a0],r1l
; Read user ID.
mov.w r2,a0
No (no user ID matched)
temp-1[a0] == r1l
Yes
No
(r2 = r2 - 1) == 0
; 7-byte confirmation completed?
Yes (confirmation completed)
bset srd11
; Set to ID identified.
command_handler
ID_code_addr:
;
.lword 0efdfh ; ID1
.lword 0efe3h ; ID2
.lword 0efebh ; ID3
.lword 0efefh ; ID4
.lword 0eff3h ; ID5
.lword 0eff7h ; ID6
.lword 0effbh ; ID7

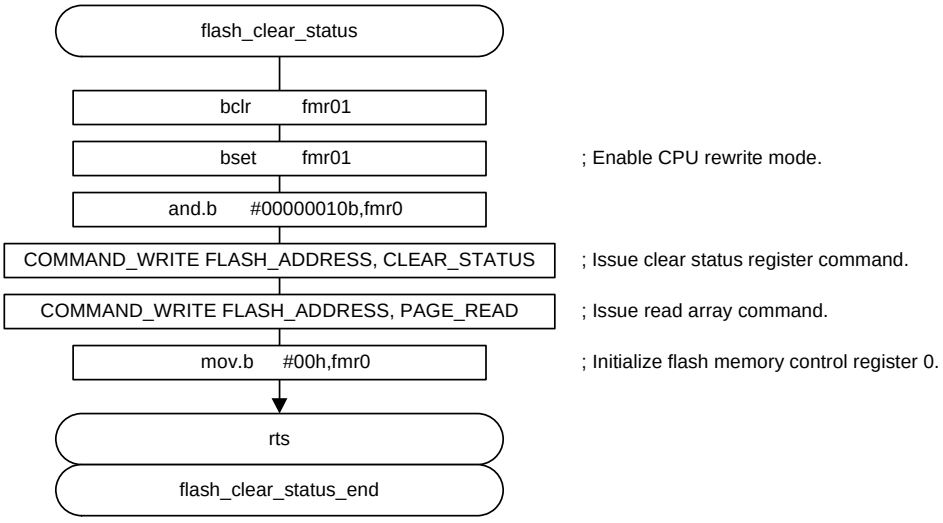
(13) Version output function



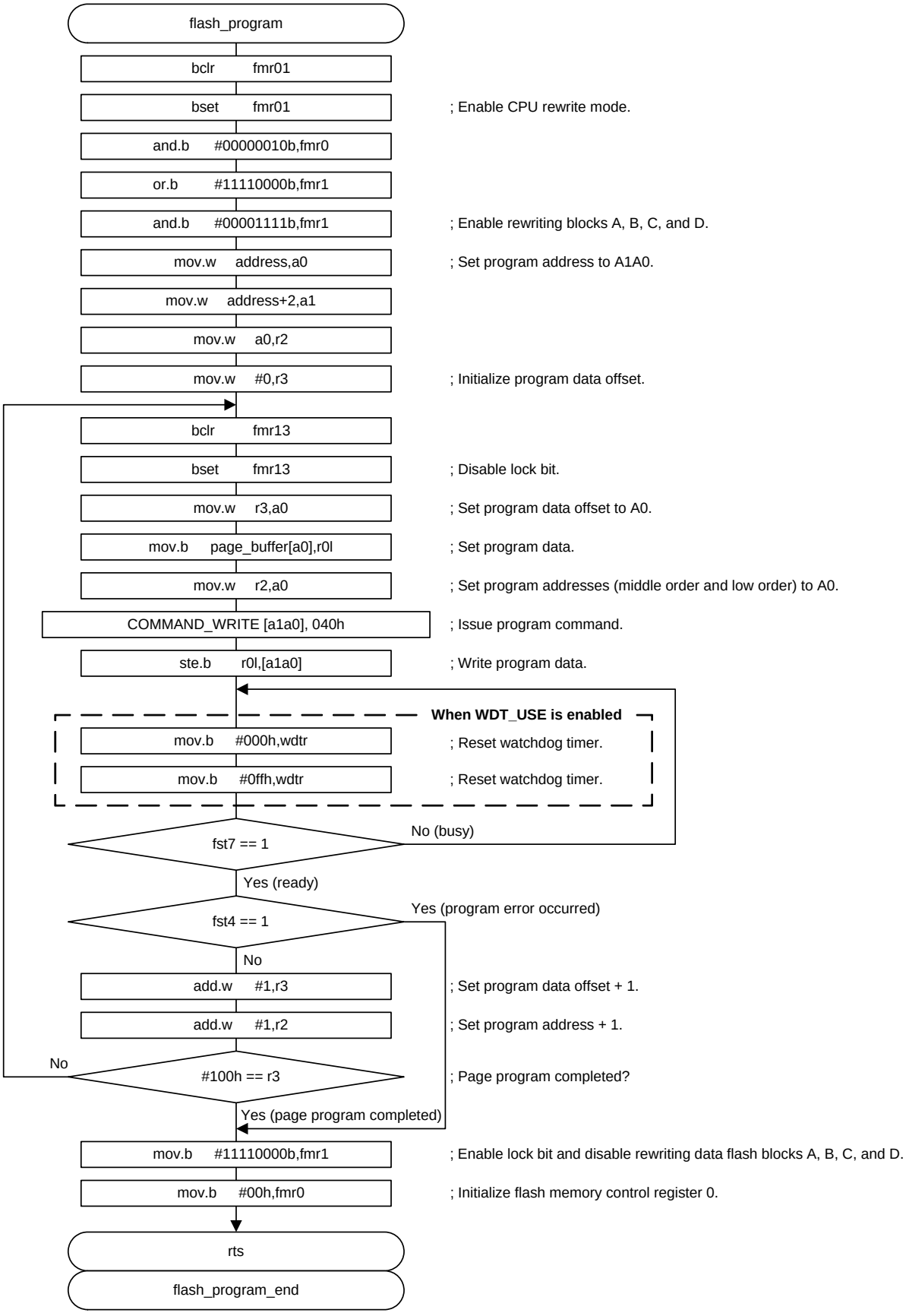
(14) RAM execute routine



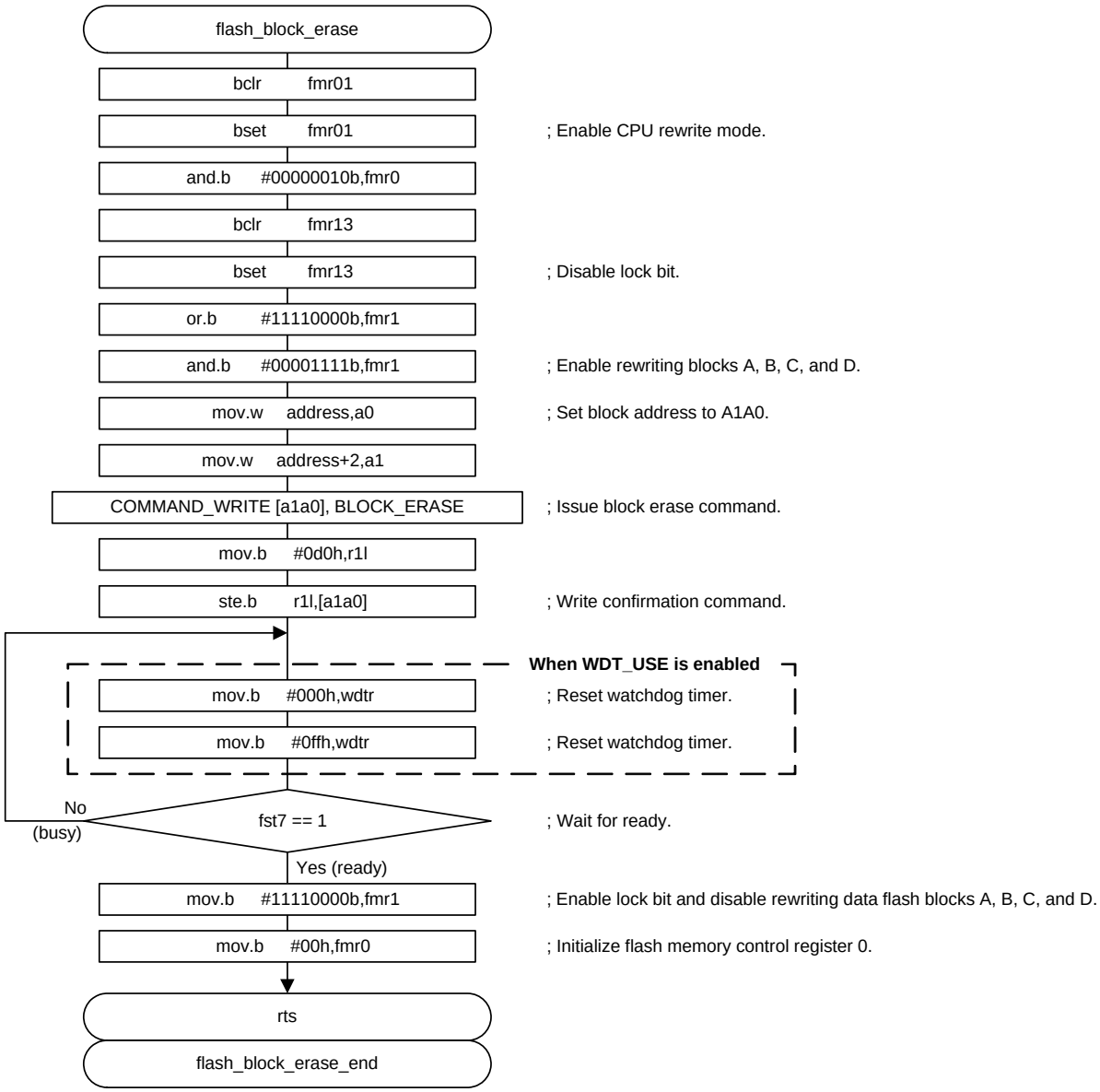
(15) Clear status register to flash memory (execute in RAM)



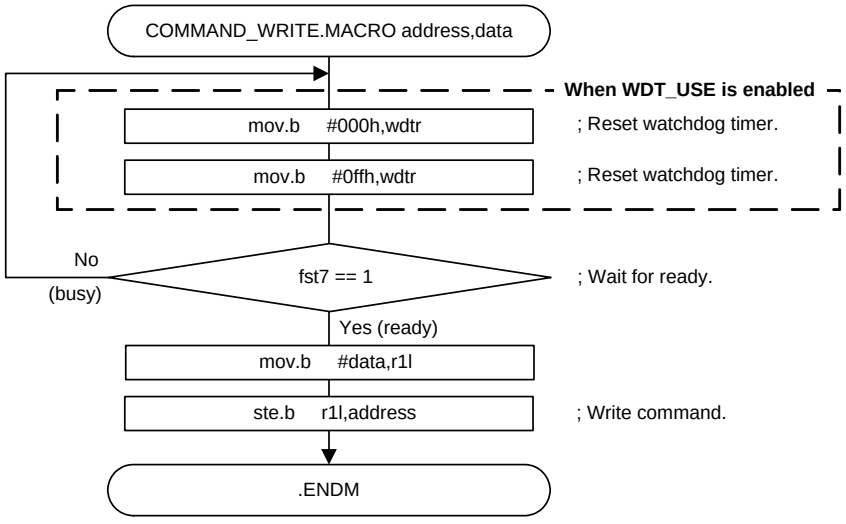
(16) Page program to flash memory (execute in RAM)



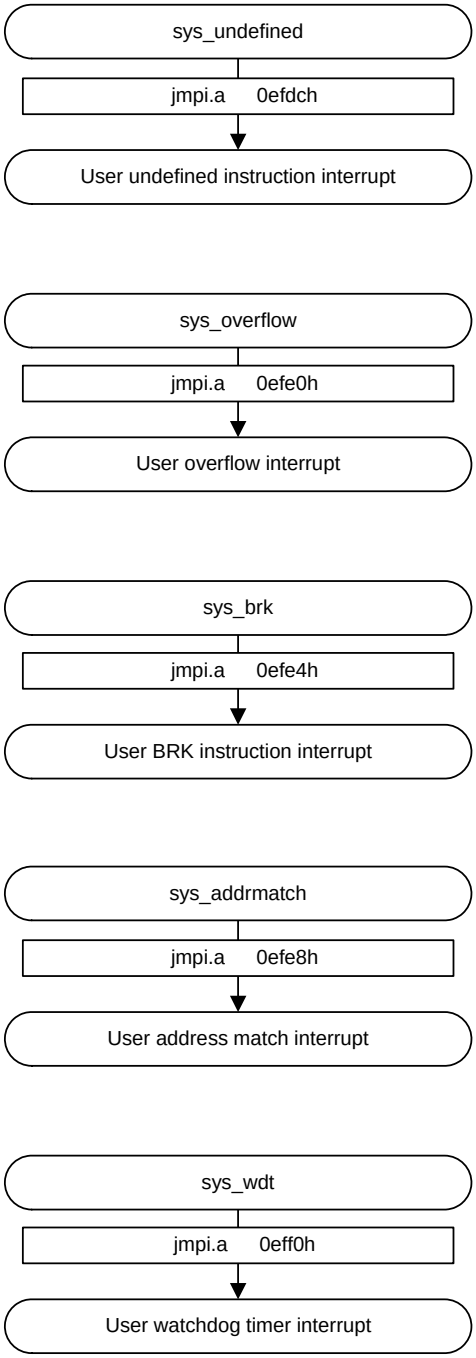
(17) Block erase to flash memory (execute in RAM)



(18) Command write macro



(19) System interrupt



4. Downloader Communication Protocol

4.1 Commands

4.1.1 Control Command List

Control commands are listed below.

Control commands	1 Byte	2 Bytes	3 Bytes	4 Bytes	5 Bytes	6 Bytes	7 Bytes or More	ID Unchecked
Page read	FFH	Middle-order address	High-order address	Data	Data	Data	Up to data	Not acceptable
Page program	41H	Middle-order address	High-order address	Data	Data	Data	Up to data	Not acceptable
Block erase	20H	Middle-order address	High-order address	D0H				Not acceptable
Read status register	70H	SRD	SRD1					Acceptable
Clear status register	50H							Not acceptable
ID check function	F5H	ID1	ID2	ID3	ID4	ID5	Up to ID7	Acceptable
Version information output function	FBH	Program downloader version		User version				Acceptable

SRD: Status register data
SRD1: Status register data 1

Notes:
1. The shadowed areas show a transfer from the MCU (program downloader) to a programmer, the rest show a transfer from a programmer to the MCU (program downloader).
2. User program area blank product IDs are identified and all commands can be accepted.
3. The number of receive data is not checked and the timeout error is not processed in the downloader. When transmitting a command, make sure there is no excess or shortage of data.

4.2 Page Read

4.2.1 Operation

The page read command reads the specified user ROM area in the flash memory in units of 256 bytes. Specify the area to be read by the high-order addresses (A16 to A23) and middle-order addresses (A8 to A15). The target bytes are the 256 bytes from addresses xxxx00h to xxxxFFh.

4.2.2 Packet

	1st byte	2nd byte	3rd byte	4th byte	Up to 259th byte
	Command	Address		Data	Up to data
Programmer to MCU	FFh	Middle-order address	High-order address		
MCU to Programmer				Data 0	Up to Data 255

Data 0: Low-order address is 00h
Data 255: Low-order address is FFh

4.2.3 Procedure

(1) The slave address and R/W direction bit are received after receiving the start condition.
(2) Page read command FFh is received at the first byte.
(3) The middle-order address is received at the second byte and the high-order address is received at the third byte.
(4) The slave address and R/W direction bit are received after receiving the restart condition.
(5) The content in low-order address 00h is sequentially transmitted from the fourth byte.
(6) The start condition is received after transmitting the last data.

When NACK is detected after transmitting the last data, the SCL line is released and the MCU waits for the stop condition. When ACK is detected, the dummy data (FFh) continues being transmitted until NACK is detected.

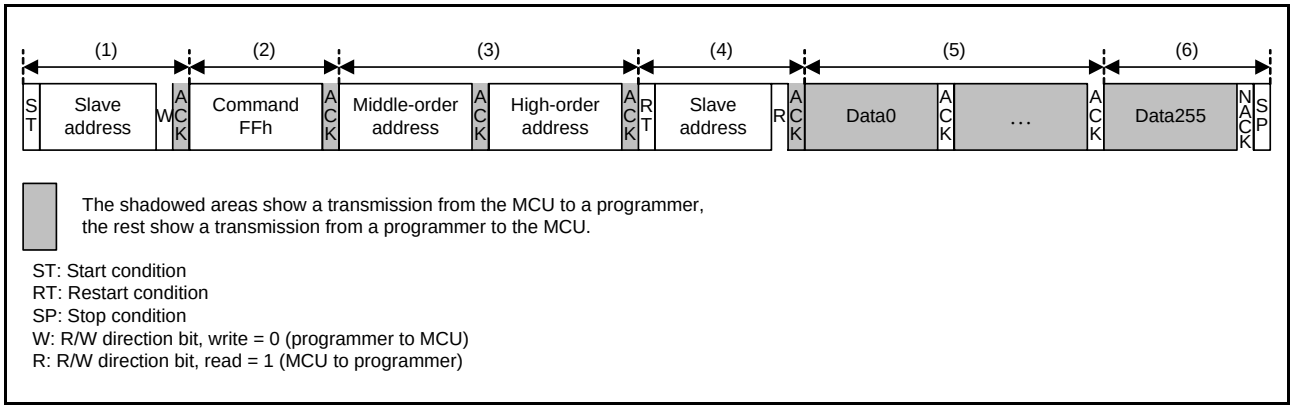


Figure 4.1 Page Read

4.3 Page Program

4.3.1 Operation

The page program command programs the data to the specified user ROM area in the flash memory in units of 256 bytes. Specify the area to be programmed by the high-order addresses (A16 to A23) and middle-order addresses (A8 to A15). The target bytes are the 256 bytes from addresses xxxx00h to xxxxFFh.

4.3.2 Packet

	1st byte	2nd byte	3rd byte	4th byte	Up to 259th byte
	Command	Address		Data	Up to data
Programmer to MCU	41h	Middle-order address	High-order address	Data 0	Up to Data 255
MCU to Programmer					

Data 0: Low-order address is 00h
Data 255: Low-order address is FFh

4.3.3 Procedure

(1) The slave address and R/W direction bit are received after receiving the start condition.
(2) Page program command 41h is received at the first byte.
(3) The middle-order address is received at the second byte and the high-order address is received at the third byte.
(4) The programming data to low-order address 00h is received from the fourth byte.
(5) The stop condition is received after receiving the last data.

When the stop condition is detected after receiving 256-byte programming data, programming starts. When the programming data is less than 256 bytes, transmit FFh for the shortage. When the programming data is less than 256 bytes and the stop condition is detected, the received commands are presumed to be command errors and programming is not performed. When the programming data is more than 257 bytes, the MCU waits for the stop condition while write data from the 257the byte is being discarded. When the stop condition is detected, a write operation of the 256 bytes already received starts. If an error occurs during programming, the SR4 bit becomes 1 (program status ends in error).

After executing this command, confirm the status of the flash memory with the read status register command.

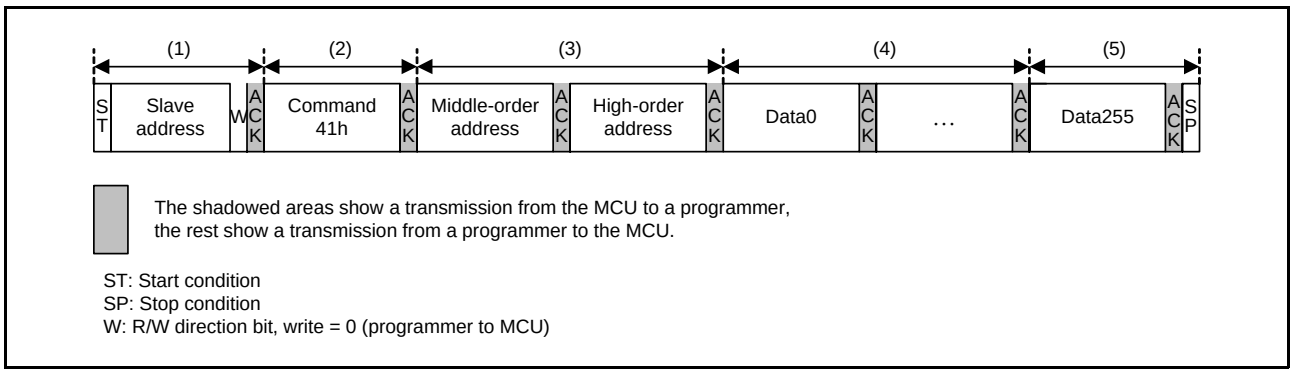


Figure 4.2 Page Program

4.4 Block Erase

4.4.1 Operation

The block erase command erases a specified block area in the user ROM area of the flash memory. Specify a block area by the eight high-order bits (A16 to A23) and eight middle-order bits (A8 to A15) at a given address of the block to be erased.

4.4.2 Packet

	1st byte	2nd byte	3rd byte	4th byte	Up to 259th byte
	Command	Block address		Confirmation command	
Programmer to MCU	20h	Middle-order address	High-order address	D0h	
MCU to Programmer					

4.4.3 Procedure

(1) The slave address and R/W direction bit are received after receiving the start condition.
(2) Block erase command 20h is received at the first byte.
(3) The middle-order address is received at the second byte and the high-order address is received at the third byte.
(4) Confirmation command D0h is received at the fourth byte.
(5) The stop condition is received.

When the stop condition is detected after receiving confirmation command D0h, erasing to the specified block starts. The erase operation sets the contents of the flash memory to FFh. If an error occurs, the SR5 bit becomes 1 (erase status ends in error).

After executing this command, confirm the status of the flash memory with the read status register command.

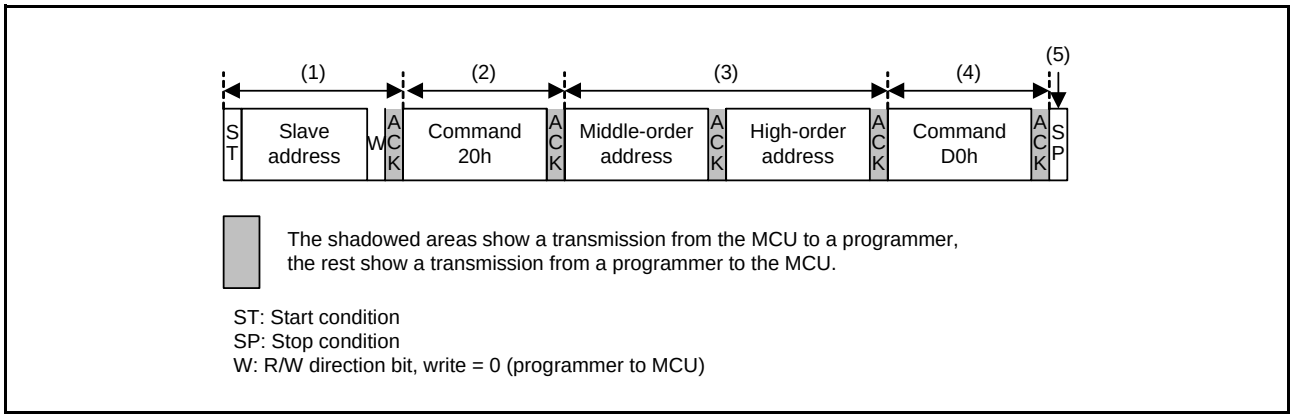


Figure 4.3 Block Erase

4.5 Read Status Register

4.5.1 Operation

The read status register command confirms the operating status of the flash memory.

4.5.2 Packet

	1st byte	2nd byte	3rd byte	4th byte	Up to 259th byte
	Command	SRD			
Programmer to MCU	70h				
MCU to Programmer		SRD output	SRD1 output		

SRD: Status register data
SRD1: Status register data 1

4.5.3 Procedure

(1) The slave address and R/W direction bit are received after receiving the start condition.
(2) Read status register command 70h is received at the first byte.
(3) The slave address and R/W direction bit are received after receiving the restart condition.
(4) SRD is transmitted at the second byte.
(5) SRD1 is transmitted at the third byte.
(6) The stop condition is received.

When NACK is detected after transmitting SRD1, the SCL line is released and the MCU waits for the stop condition. When ACK is detected, the dummy data (FFh) continues being transmitted until NACK is detected.

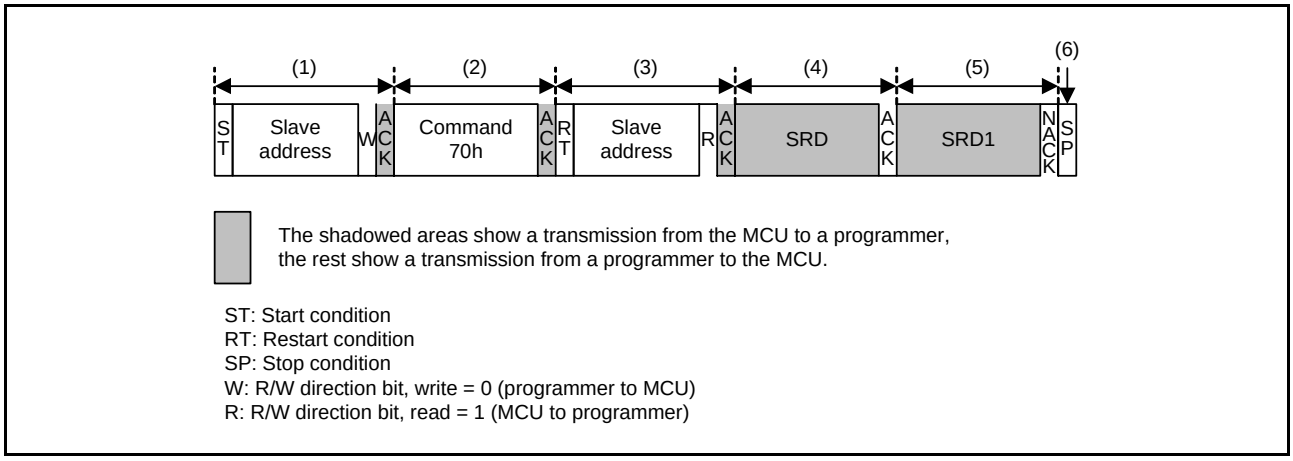


Figure 4.4 Read Status Register

4.5.4 SRD Register

Each Bit of SRD	Status Name	Definition	
		1	0
SR7 (bit 7)	Sequencer status	Ready	Busy
SR6 (bit 6)	Reserved		
SR5 (bit 5)	Erase status	Error	Completed normally
SR4 (bit 4)	Program status	Error	Completed normally
SR3 (bit 3)	Reserved		
SR2 (bit 2)	Reserved		
SR1 (bit 1)	Reserved		
SR0 (bit 0)	Reserved		

(1) Sequencer status
The sequencer status shows the operating status of the flash memory. This bit becomes 0 (busy) during auto-programming or auto-erasing. This bit becomes 1 (ready) during auto-programming or auto-erasing.

(2) Erase status
The erase status shows the erase operating status. If an error occurs, this bit becomes 1. This bit is set to 0 when the clear status register command is executed.

(3) Program status
The program status shows the programming status. If an error occurs, this bit becomes 1. This bit is set to 0 when the clear status register command is executed.

Both bits SR5 and SR4 become 1 in the following cases:
- The defined command is not written correctly.
- Data other than values which can be written to the second bus cycle data of the block erase command (D0h or FFh) is written in the cycle to input the block erase confirmation command. When FFh is written, the MCU enters read array mode and the command is canceled.

(4) Reserved bit
When read, the content is undefined.

4.5.5 SRD1 Register

Each Bit of SRD1	Status Name	Definition	
		1	0
SR15 (bit 7)	Reserved		
SR14 (bit 6)	Reserved		
SR13 (bit 5)	Reserved		
SR12 (bit 4)	Reserved		
SR11 (bit 3)	ID check	00: Not checked 01: ID Not matched 10: Reserved 11: Checked	
SR10 (bit 2)			
SR9 (bit 1)	Reserved		
SR8 (bit 0)	Reserved		

(1) ID check
These bits indicate the ID check results.

(2) Reserved bit
When read, the content is undefined.

4.6 Clear Status Register

4.6.1 Operation

The clear status register command initializes a status register. Initialize the status register before executing the erase or the page program to the flash memory.

4.6.2 Packet

	1st byte	2nd byte	3rd byte	4th byte	Up to 259th byte
	Command				
Programmer to MCU	50h				
MCU to Programmer					

4.6.3 Procedure

(1) The slave address and R/W direction bit are received after receiving the start condition.
(2) The clear status register command 50h is received at the first byte.
(3) The stop condition is received.

When the stop condition is detected after receiving the clear status register command 50h, the status register is initialized.

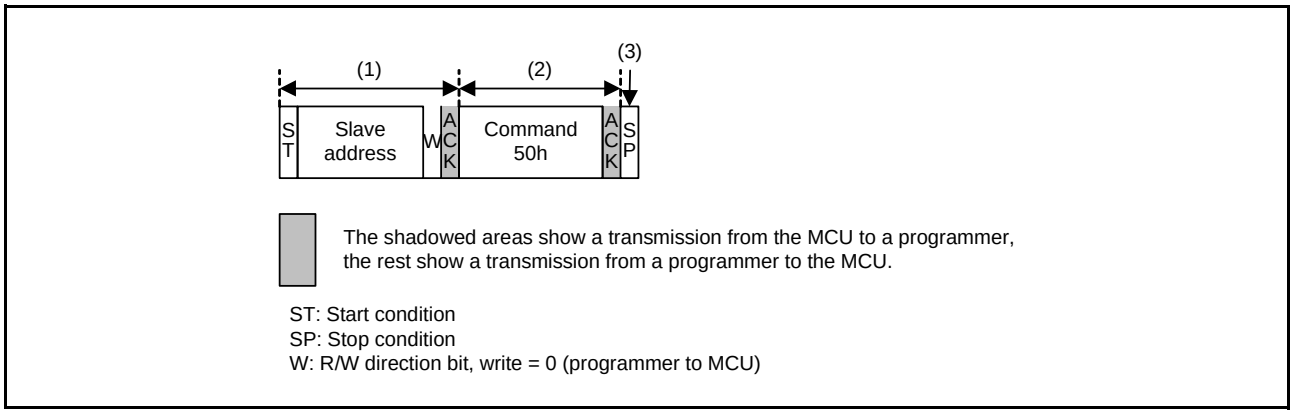


Figure 4.5 Clear Status Register

4.7 ID Check Function

4.7.1 Operation

This function compares the ID received from the programmer and the user ID code stored in the virtual fixed vector address. The ID check results are stored in SR11 to SR10 in the SRD1 register.

4.7.2 Packet

	1st Byte	2nd Byte	3rd Byte	4th Byte	Up to 8th Byte
	Command	ID			
Programmer to MCU	F5h	ID1	ID2	ID3	Up to ID7
MCU to Programmer					

4.7.3 Procedure

(1) The slave address and R/W direction bit are received after receiving the start condition.
(2) ID check function command F5h is received at the first byte.
(3) ID1 to ID7 are received from the second byte to the eighth byte, respectively.
(4) The stop condition is received after receiving ID7.

When the stop condition is detected after receiving ID7, the ID check starts. However, a user program area blank product returns to waiting for the start condition state without performing ID check. When ID1 to ID7 all match, bits SR11 to SR10 become 11b (verified). If any of the IDs do not match, bits SR11 to SR10 become 01b (verify not matched).

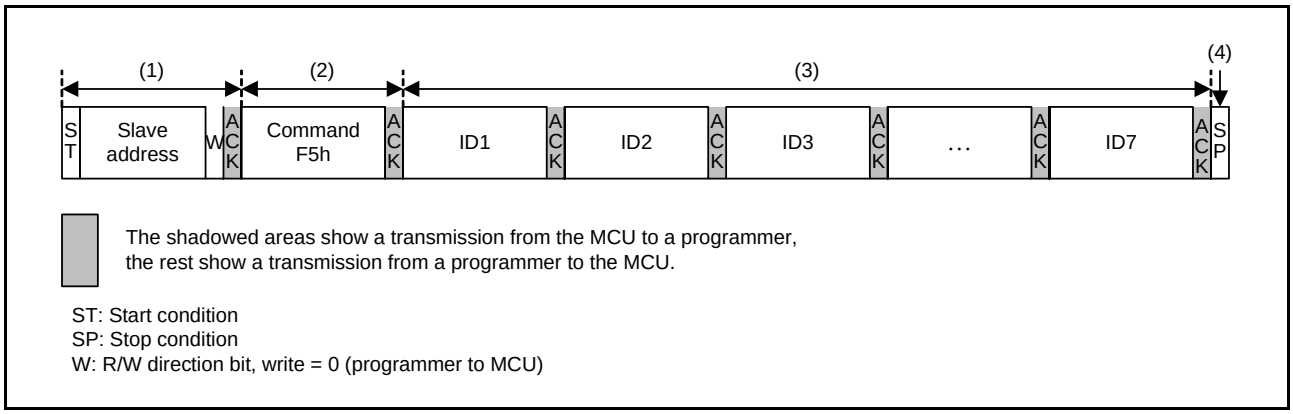


Figure 4.6 ID Check Function

4.8 Version Information Output Function

4.8.1 Operation

This function transmits version information of the program downloader and user program.

4.8.2 Packet

	1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	Up to 259th Byte
	Command	Version				
Programmer to MCU	F8h					
MCU to Programmer		Program downloader		User		

4.8.3 Procedure

(1) The slave address and R/W direction bit are received after receiving the start condition.
(2) The version information output function command FBh is received at the first byte.
(3) The slave address and R/W direction bit are received after receiving the restart condition.
(4) The program downloader version is transmitted at the high-order second byte first and then the low-order third byte.
(5) The user program version is transmitted at the high-order fourth byte first and then low-order fifth byte.
(6) The stop condition is received.

When NACK is detected after transmitting the user program version, the SCL line is released and the MCU waits for the stop condition. When ACK is detected, the dummy data (FFh) continues being transmitted until NACK is detected.

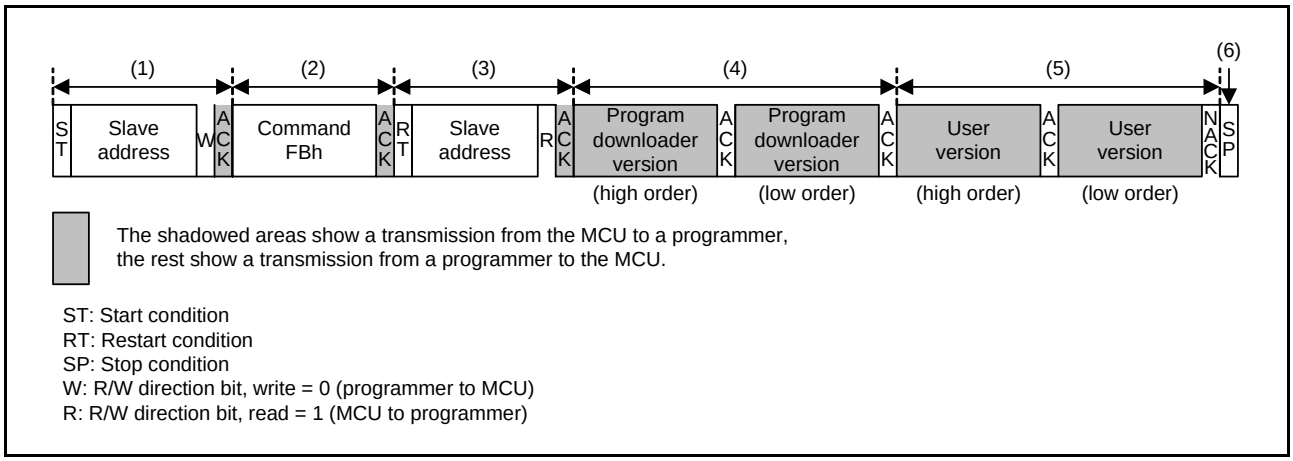


Figure 4.7 Version Information Output Function

4.8.4 Version Data

For the example shown below, the program download version is transmitted after 01h is set to the high order and 00h to the low order of the program downloader version, and 00h is set to the high order and 10h to the low order of the user version.

When the program downloader version is Ver.1.00 and the user version is Ver.0.10:

Program downloader version data
(in the bt_r835a.a30 file)

```
  .org        version_string

  .word       0100h                ; Program Downloader version (Ver.1.00)
```

User version data
(in the sect30.inc file for 6. User Program Example)

```
User_Ver   .equ        0010h     ; User version (Ver.0.10)
```

5. Error Handling

5.1 Command Error

When commands other than control commands shown in 4.1.1 Control Command List are received, the received commands are presumed to be command errors and a dummy read of the I2C receive buffer or transmission of dummy data (FFh) repeats until the stop condition is detected. When the stop condition is detected, the MCU returns to waiting for the start condition state.

5.2 Stop Condition Detection Error

When the stop condition is detected in a communication, the command in this communication is disregarded and the MCU returns to waiting for the start condition state.

5.3 Data Excess or Shortage

When the received data exceeds the number of bytes selected by the command, the MCU waits for the stop condition while reading and discarding the excess data. When the stop condition is detected, command handling is performed using the data initially received. When the data received is less than the selected number of bytes, the command is disregarded and command handling is not performed.

When ACK is detected and data continues being transmitted after transmitting the number of bytes data selected by the command, dummy data (FFh) continues being transmitted until NACK is detected. When NACK is detected, the SCL line is released and the MCU waits for the stop condition. When the stop condition is detected, the MCU returns to waiting for the start condition state.

6. User Program Example

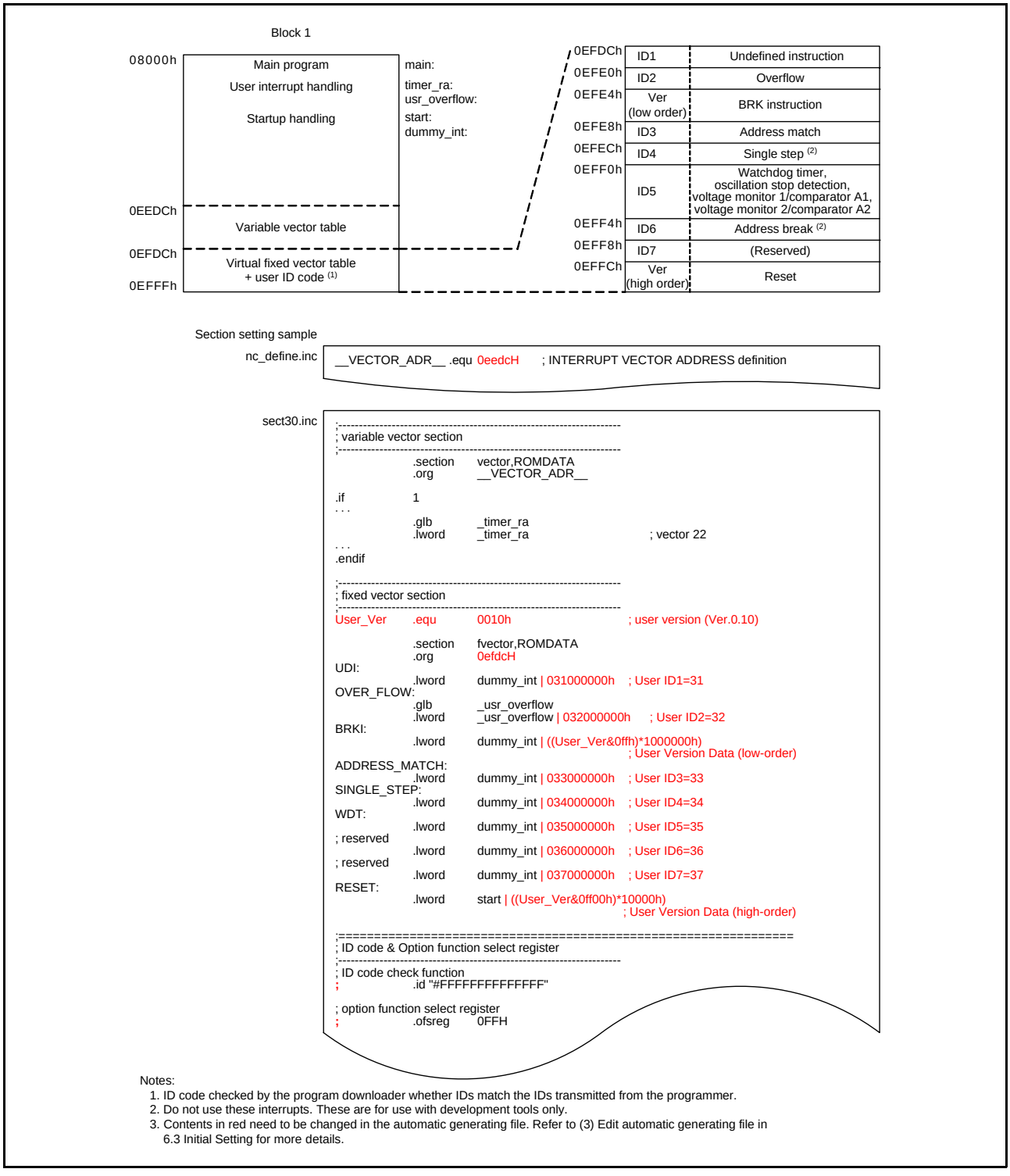
The program downloader rewrites the user programs other than the user program in block 0 according to the programmer. An example of the user program is shown below.

6.1 Function

The LEDs connected to I/O ports P3_1, P3_3, P3_4, and P3_6 light.

6.2 Memory Map

Figure 6.1 shows a User Program Memory Map.



Notes:
1. ID code checked by the program downloader whether IDs match the IDs transmitted from the programmer.
2. Do not use these interrupts. These are for use with development tools only.
3. Contents in red need to be changed in the automatic generating file. Refer to (3) Edit automatic generating file in 6.3 Initial Setting for more details.

Figure 6.1 User Program Memory Map

6.3 Initial Settings

(1) Vector table

Allocate the virtual fixed vector table to block 1 to use an interrupt by the user program.

(2) ID code

Set an ID code in the virtual fixed vector table. Do not opt to generate an ID code file when compiling.

(3) Edit automatic generating file

When the project type is made in the Application and the initial setting file is automatically generated by the High-performance Embedded Workshop (HEW), change the sect30.inc file and nc_define.inc file as follows (see Figure 6.1):

- Change the allocation address of the locatable table to 0EEDCh, and the virtual fixed vector table to 0EFDCh.
- Set an additional ID code to the virtual fixed vector table.
- Add the symbol definition of the user version data and user version data setting to the virtual fixed vector table.
- Comment out the assembler expansion function direction instructions ".ID" (set an ID code) and ".OFSREG" (set a value to the OFS register).

7. Programmer Example

7.1 Control Pins

(1) Pins SCL and SDA

These pins are for transmitting and receiving in I2C-bus.

(2) P3_3 pin

This pin is used to select the program downloader or the user program.

(3) $\overline{\text{RESET}}$ pin

This pin controls MCU resets from the programmer.

(4) Pins VCC and VSS

Adjust high level from the programmer to the MCU's VCC level and low level from the programmer to the MCU's VSS level, respectively.

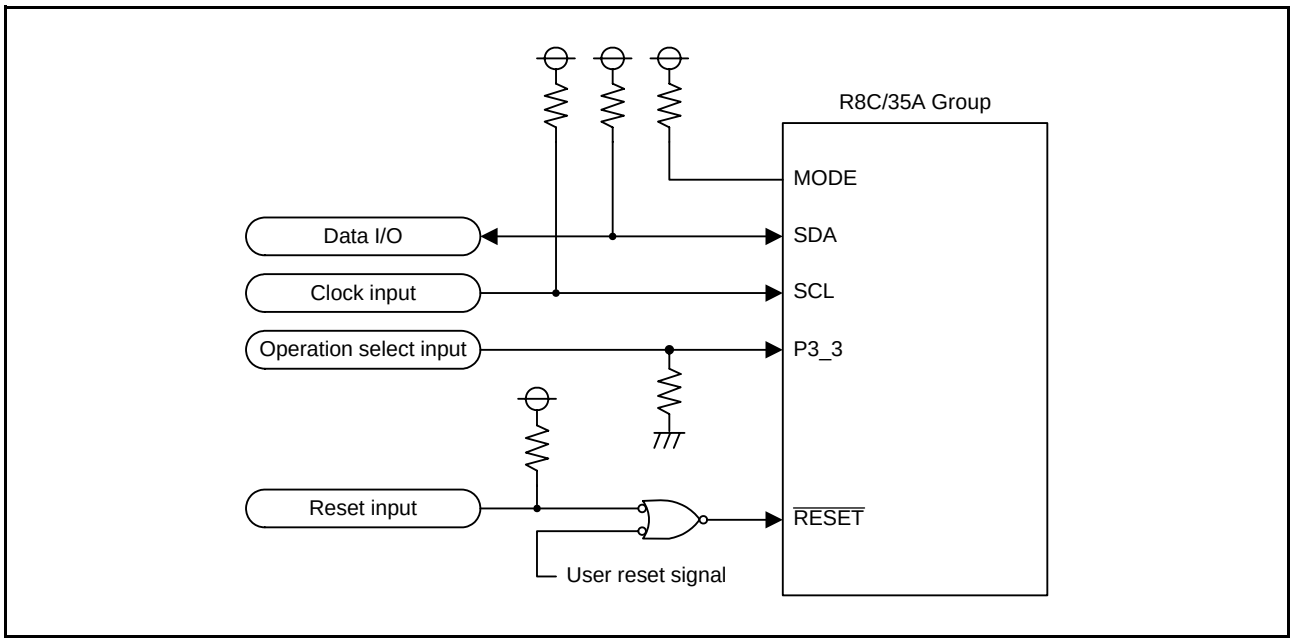


Figure 7.1 Programmer Configuration

7.2 Programmer Communication Protocol

- I2C-bus master mode is used to communicate with the program downloader.
- When the restart condition is transmitted for the read command, the R/W direction bit is switched as follows:
- Set the R/W direction bit to 0 (programmer to MCU) when the slave address is transmitted after transmitting the start condition.
- Set the R/W direction bit to 1 (MCU to programmer) when the slave address is transmitted after transmitting the restart condition.

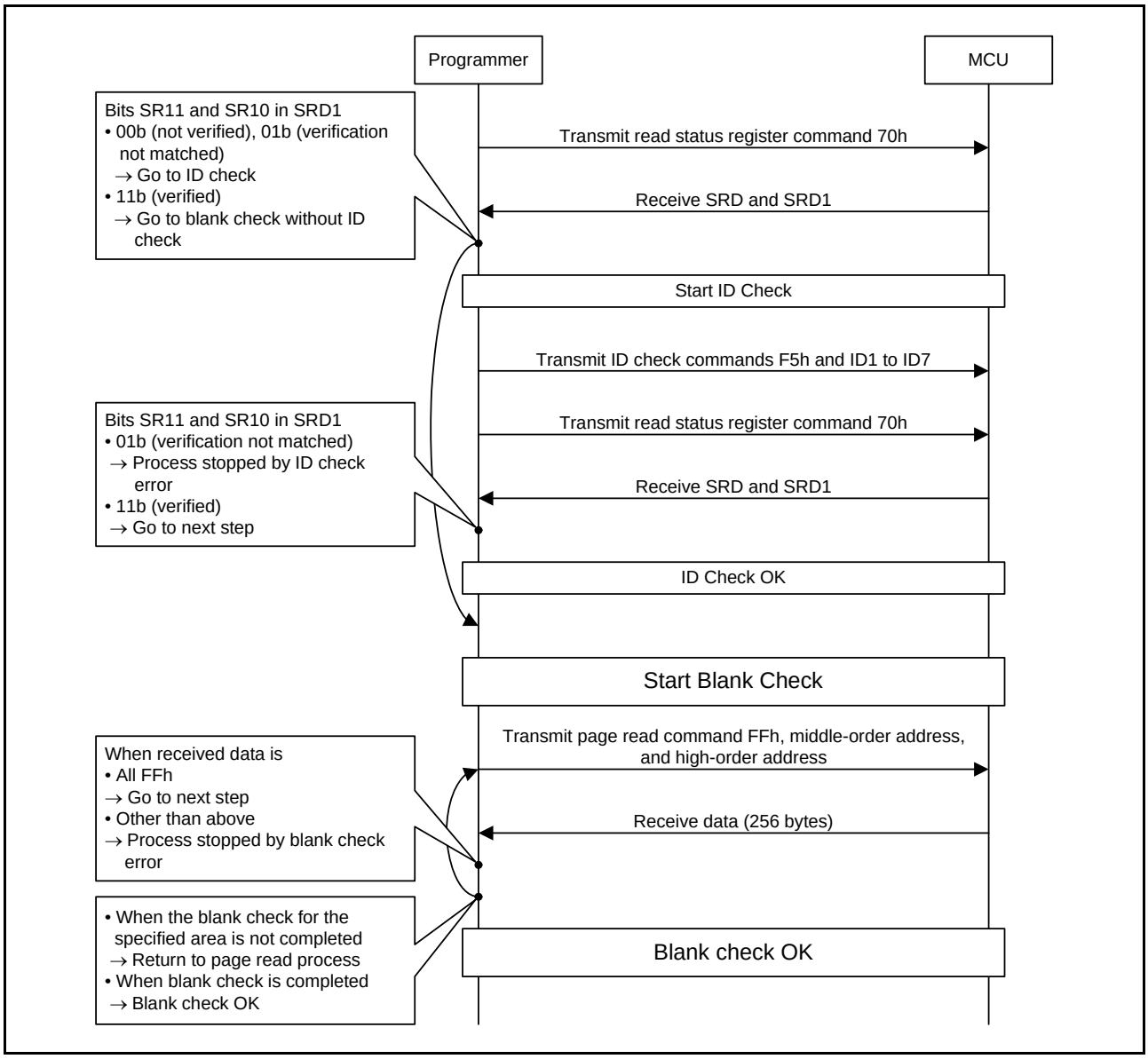
7.3 Programmer Functions

The following are the functions necessary for the programmer:

- Blank Check
- Erase
- Program
- Verify
- Read

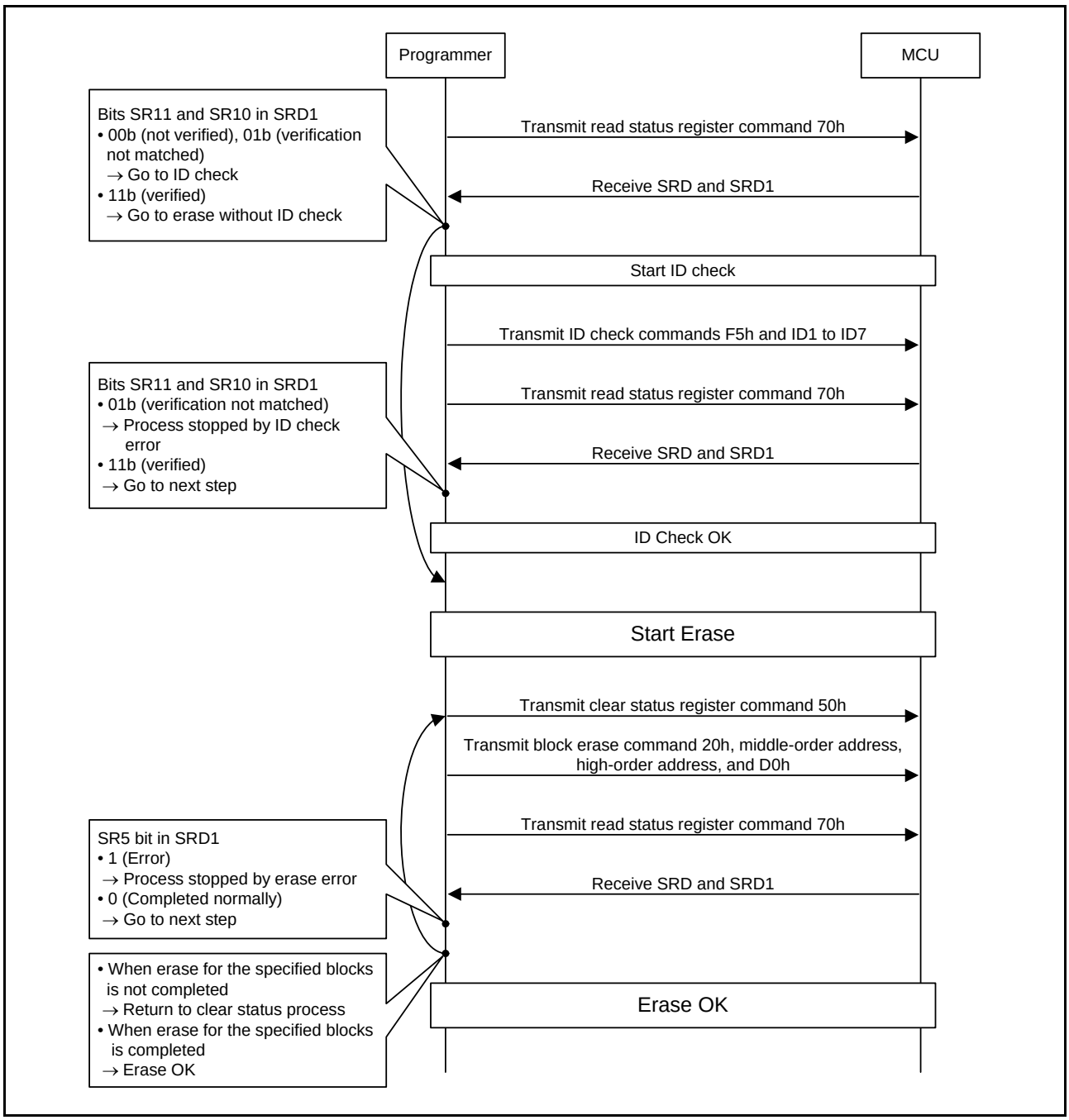
7.4 Blank Check

Data (program) in the specified area automatically or manually is read from the MCU's on-chip flash memory. The programmer confirms that all read data is blank (FFh).



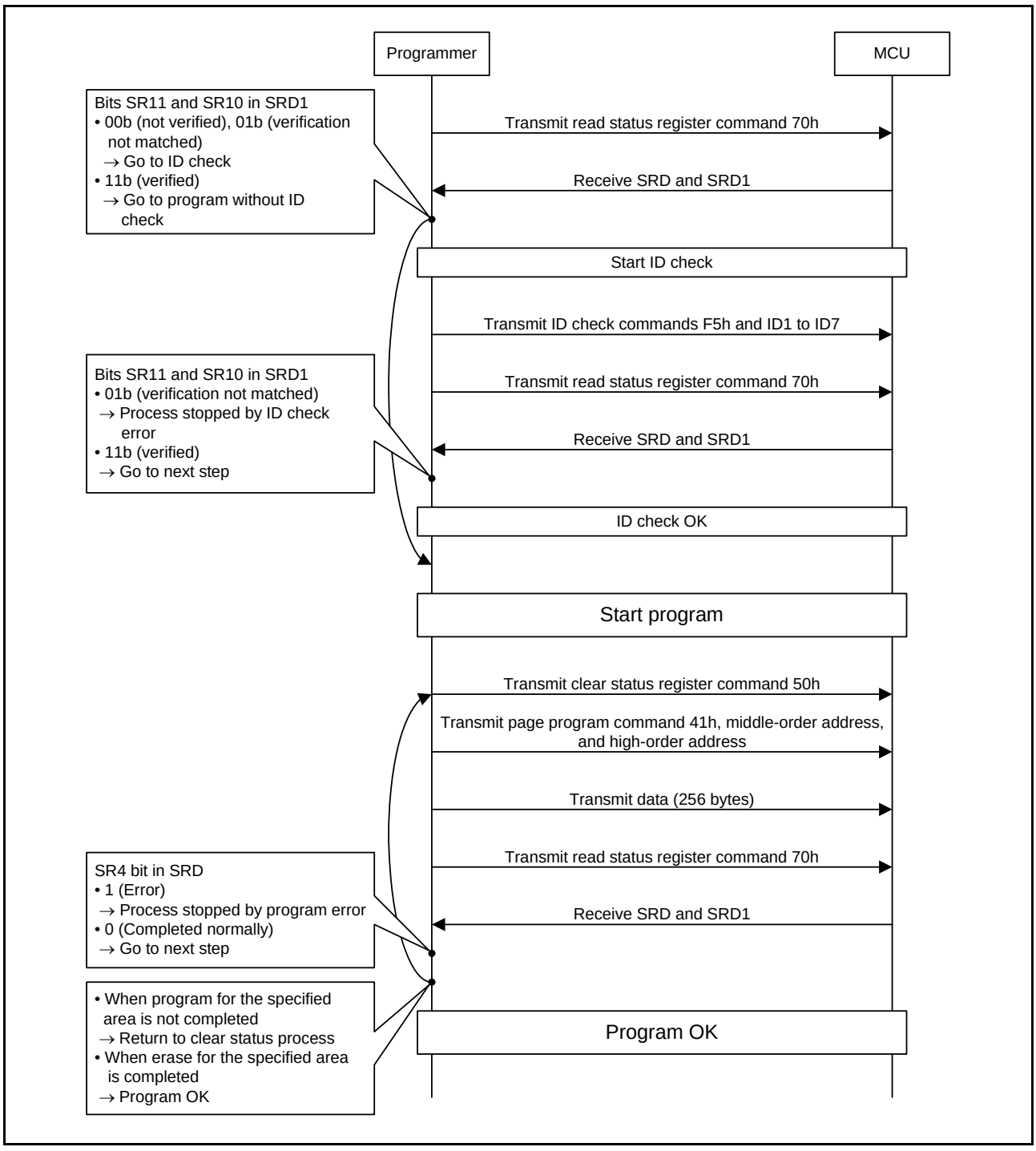
7.5 Erase

Data (program) in the MCU's specified on-chip flash memory blocks automatically or manually is erased.



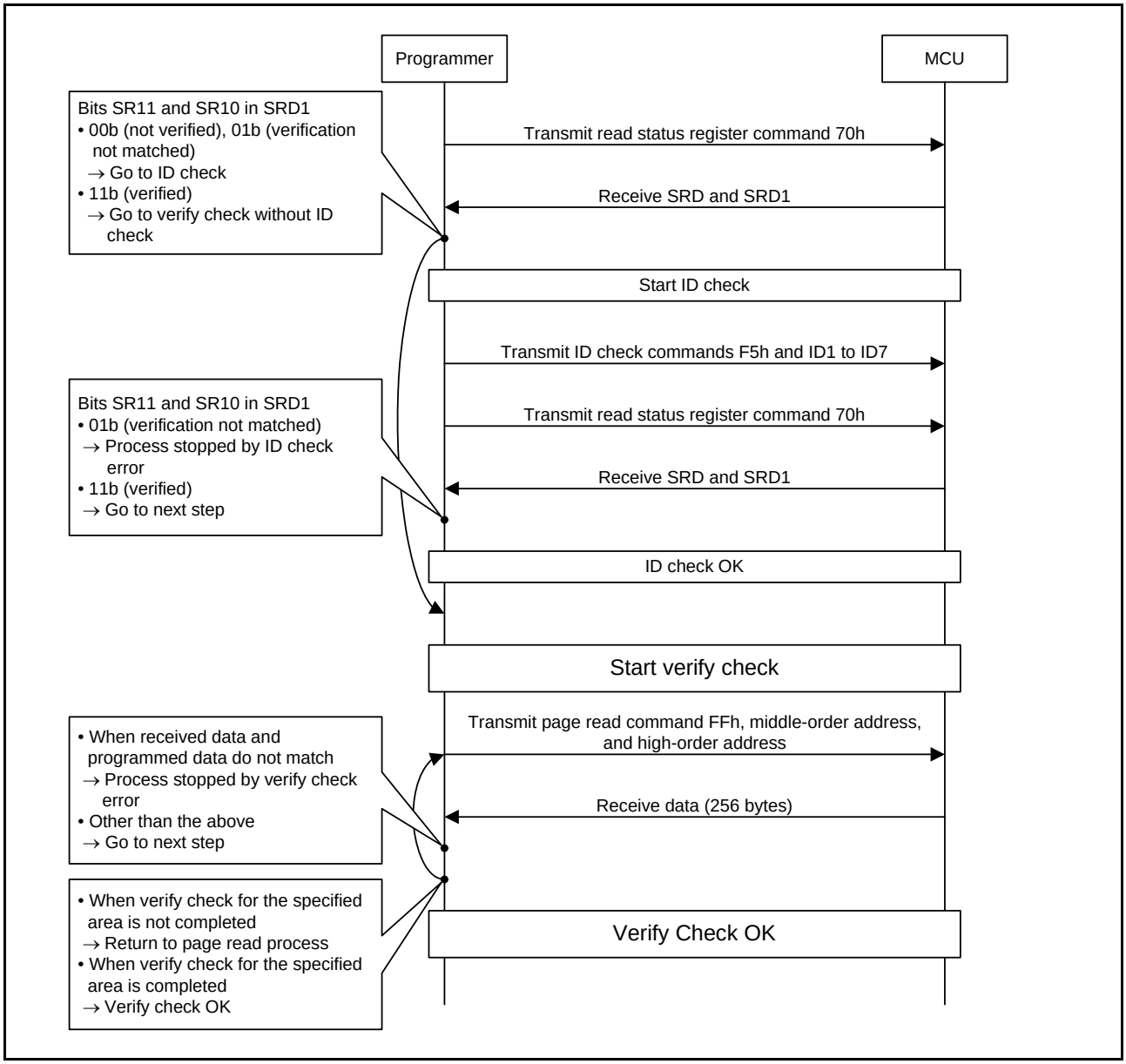
7.6 Program

Data (program) in the MCU's specified on-chip flash memory area automatically or manually is programmed.



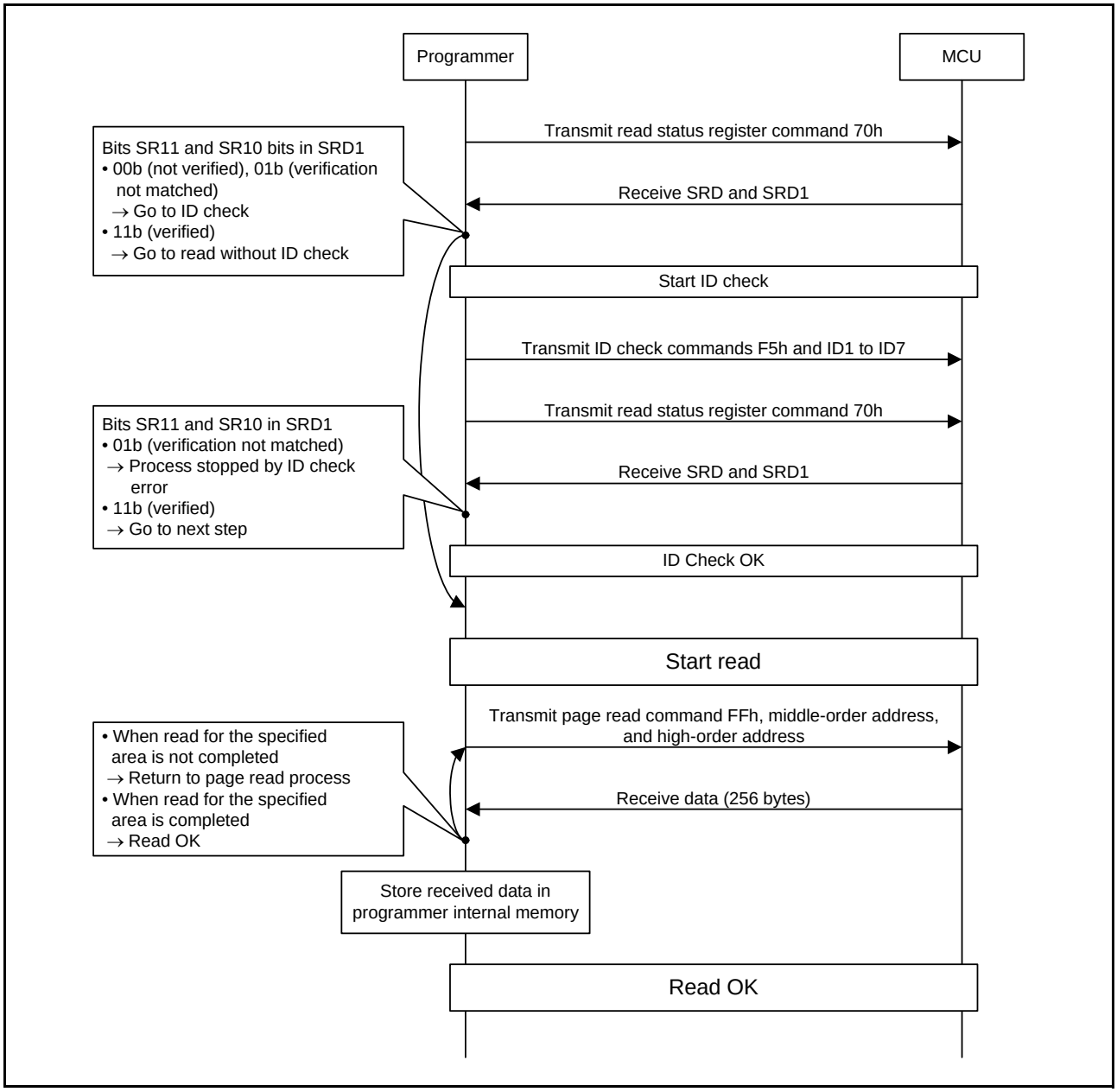
7.7 Verify

Data (program) in the specified area automatically or manually is read from the MCU's on-chip flash memory. The programmer compares the read data with the memory data (program) in the programmer to confirm that they match.



7.8 Read

This function allows reading the data (program) in the automatically or manually specified area from the MCU with on-chip flash memory.
The programmer stores the read data in its internal memory



8. Sample Programming Code

A sample program can be downloaded from the Renesas Technology website.
To download, click “Application Notes” in the left-hand side menu on the R8C/35A Group page.

9. Reference Documents

Hardware Manual
R8C/35A Group Hardware Manual Rev.0.40
The latest version can be downloaded from the Renesas Technology website.

Technical News/Technical Update
The latest information can be downloaded from the Renesas Technology website.

Website and Support

Renesas Technology website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

REVISION HISTORY	R8C/35A Group I2C Version Program Downloader

Rev.	Date	Description	
		Page	Summary
1.00	Dec 29, 2009	–	First Edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.