Date: Feb. 24, 2012

RENESAS TECHNICAL UPDATE

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

Product Category	MPU & MCU		Document No.	TN-16C-A207A/E	Rev.	1.00
Title	Errata to R32C/121 Group Hardware Manual		Information Category	Technical Notification		
Applicable Product	R32C/121 Group	Lot No.	Reference Document	R32C/121 Group Hardware Manual Rev. 1.10 (REJ09B0477-0110)		

This document describes corrections to the R32C/121 Group Hardware Manual, Rev. 1.10.
The corrections are indicated in red in the list below.

- Page 11 of 604, description “Output of the clock with the same frequency as fC, f8, or f32” in the description for the Clock output in Table 1.7 is corrected as follows:
  “Output of the clock with the same frequency as low speed clocks, f8, or f32”

- Page 15 of 604, register symbol “R3R0” in line 3 of 2.1.1 is corrected as follows:
  “R3R1”

- Page 24 of 604, description of register name “Group 1 Timer Measurement Prescaler Register 6/7” in Table 4.6 is corrected as follows:
  “Group 1 Time Measurement Prescaler Register 6/7”

- Page 26 of 604, description of register name “Group 0 Timer Measurement Prescaler Register 6/7” in Table 4.8 is corrected as follows:
  “Group 0 Time Measurement Prescaler Register 6/7”

- Page 31 of 604, description of register name “UART2 Transmission/Receive Mode Register” in Table 4.13 is corrected as follows:
  “UART2 Transmit/Receive Mode Register”

- Page 31 of 604, description of register name “Increment/Decrement Counting Select Register” in Table 4.13 is corrected as follows:
  “Increment/Decrement Select Register”

- Page 38 of 604, reset value of the IFS0 register “X000 X000b” in Table 4.20 is corrected as follows:
  “X0X0 X000b”

- Page 42 of 604, description of register name “External Interrupt Source Select Register 0” in Table 4.24 is corrected as follows:
  “External Interrupt Request Source Select Register 0”

- Pages 57 to 58 and 71 to 72 of 604, description of register name “CAN1/0 Acceptance Mask Register 0/1/2/3/4/5/6/7” in Tables 4.39 to 4.40 and 4.53 to 4.54 is corrected as follows:
  “CAN1/0 Mask Register 0/1/2/3/4/5/6/7”

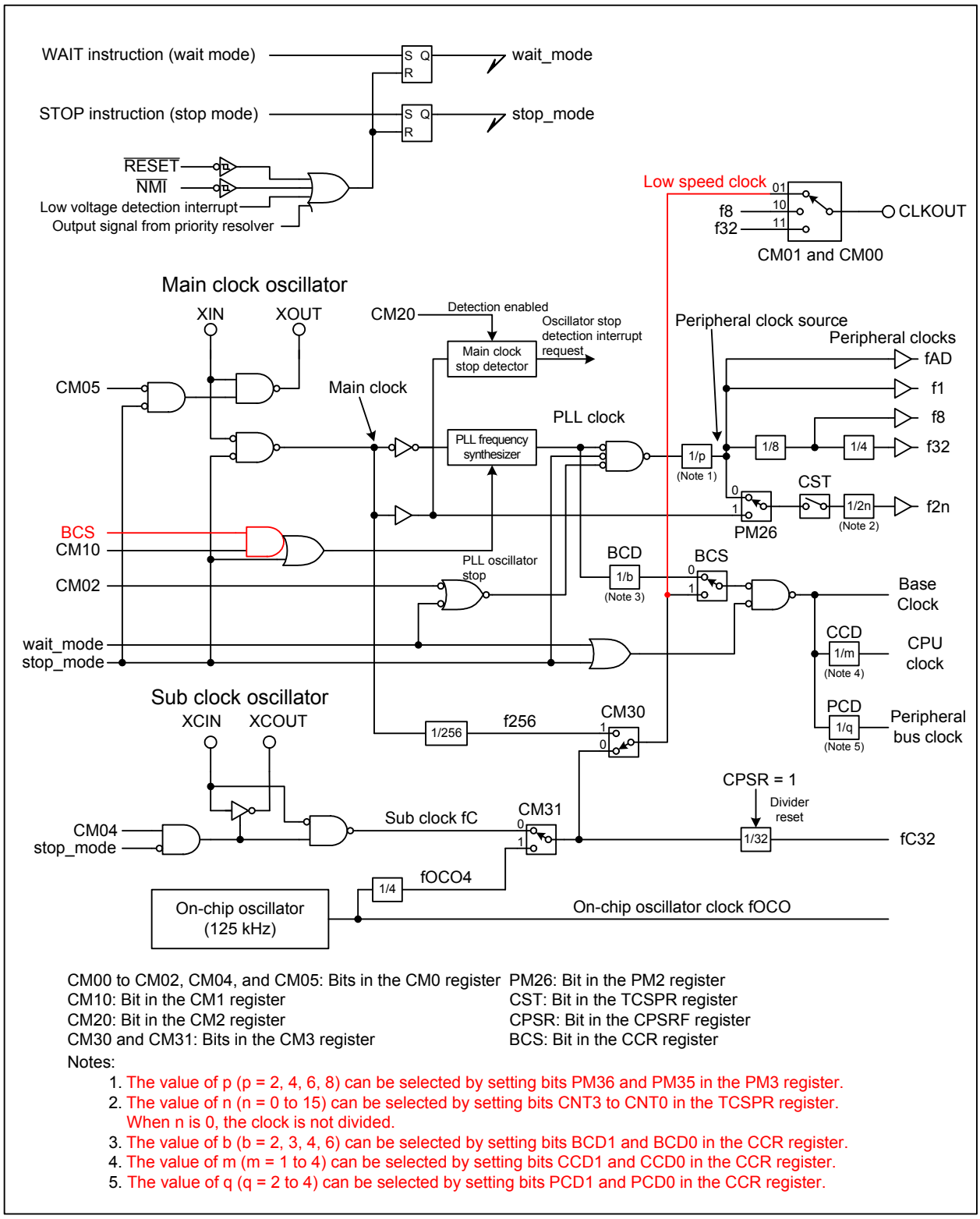
•Pages 60 and 74 of 604, descriptions of register names “CAN1/0 Reception Error Count Register” and “CAN1/0 Transmission Error Count Register” in Tables 4.42 and 4.56 are corrected as follows:
“CAN1/0 Receive Error Count Register” and “CAN1/0 Transmit Error Count Register”

•Pages 60 and 74 of 604, reset value “XXXX XX00b” for registers C1MSMR and C0MSMR in Tables 4.42 and 4.56 is corrected as follows:
“0000 0000b”

•Page 82 of 604, descriptions for the VDEN bit in Figure 6.4 are modified as follows:

Bit Symbol	Bit Name	Function	RW
VDEN	Low Voltage Detector Enable Bit	0: Low voltage detector disabled 1: Low voltage detector enabled	RW

•Page 87 of 604, Figure 7.1 is corrected as follows:



CM00 to CM02, CM04, and CM05: Bits in the CM0 register
CM10: Bit in the CM1 register
CM20: Bit in the CM2 register
CM30 and CM31: Bits in the CM3 register
PM26: Bit in the PM2 register
CST: Bit in the TCSPR register
CPSR: Bit in the CPSRF register
BCS: Bit in the CCR register

Notes:
1. The value of p (p = 2, 4, 6, 8) can be selected by setting bits PM36 and PM35 in the PM3 register.
2. The value of n (n = 0 to 15) can be selected by setting bits CNT3 to CNT0 in the TCSPR register. When n is 0, the clock is not divided.
3. The value of b (b = 2, 3, 4, 6) can be selected by setting bits BCD1 and BCD0 in the CCR register.
4. The value of m (m = 1 to 4) can be selected by setting bits CCD1 and CCD0 in the CCR register.
5. The value of q (q = 2 to 4) can be selected by setting bits PCD1 and PCD0 in the CCR register.

Figure 7.1 Clock Generation Circuitry

•Page 88 of 604, descriptions of Notes 2 and 6 in Figure 7.2 are corrected as follows:
Note 2: "The divide ratios of the base clock and peripheral bus clock should not be changed simultaneously. Doing so may cause the peripheral bus clock frequency to go over the maximum operating frequency." ("To increase the base clock frequency, the divide ratio of the peripheral bus clock should be increased before reducing the divide ratio of base clock." is deleted)
Note 6: "To use these low speed clocks, select one of them by setting bits CM31 and CM30 in the CM3 register and then set the BCS bit to 1."

•Pages 89, 103, 110, and 113 of 604, description "fC" for bits CM00 and CM01 in Figure 7.3, Section 7.6, Tables 7.3, 7.4, and 7.6 is corrected as follows:
Figure 7.3: "0 1 : Output a low speed clock"
Section 7.6: "Low speed clocks, f8, and f32 can be output from the CLKOUT pin."
Table 7.3: "Output a low speed clock"
Tables 7.4 and 7.6: "When a low speed clock is selected"

•Page 89 of 604, the following description is added to Figure 7.3 as Note 7:
"Set this bit before activating the watchdog timer. When rewriting this bit while the watchdog timer is running, set it immediately after writing to the WDTS register."

•Page 90 of 604, description of bit name "PLL Clock Oscillator Stop Bit" in Figure 7.4 is modified as follows:
"PLL Oscillator Stop Bit"

•Page 90 of 604, description of Note 2 in Figure 7.4 is corrected as follows:
"When the BCS bit in the CCR register is 0 (PLL clock selected as base clock source), the PLL frequency synthesizer does not stop oscillating even if the CM10 bit is set to 1."

•Page 90 of 604, the following description is added to Figure 7.4 as Note 4:
"This bit becomes 1 when the main clock is stopped. When setting to 0, rewrite it after the main clock oscillation is fully stabilized."

•Page 90 of 604, bit symbol "CM02" in Note 3 of Figure 7.5 is corrected as follows:
"CM20"

•Page 91 of 604, description of Note 1 in Figure 7.6 is corrected as follows:
"Rewrite this register after setting the PRC27 bit in the PRCR2 register to 1 (write enabled) and while the BCS bit in the CCR register is 0 (PLL clock)."

•Page 93 of 604, descriptions in Note 3 in Figure 7.9 are modified as follows:
"CM05 bit in the CM0 register (main clock oscillator enabled/disabled)
CM10 bit in the CM1 register (PLL oscillator enabled/disabled)"

•Page 93 of 604, the following description is added to Figure 7.9 as Note 6:
"Disable all the peripheral functions that use f2n before rewriting this bit."

•Page 94 of 604, the following description is added to Note 1 in Figure 7.10:
"Disable all the peripheral functions that use fAD, f1, f8, f32, or f2n (when the clock source is the peripheral clock source) to rewrite this register."

•Page 98 of 604, descriptions for the SEO bit in Figure 7.15 are modified as follows:

Bit Symbol	Bit Name	Function	RW
SEO	Self-oscillating Setting Bit	0: PLL lock-in 1: Self-oscillating	RW

•Page 98 of 604, the following description is added to Figure 7.16 as Note 1:
"This register is reset after setting the SEO bit in the PLC1 register to 1 (self-oscillating). Stopping the main clock or PLL prevents the register from updating."

•Page 100 of 604, description of the last paragraph in 7.1.4 is modified as follows:
"When the CSPM bit in the OFS area is 1, the on-chip oscillator clock is stopped after a reset. It starts running if the CM31 bit in the CM3 register or the PM22 bit in the PM2 register is set to 1. It is not necessary to wait for stabilization because the on-chip oscillator instantly starts oscillating."

•Page 101 of 604, description "(Refer to Figure 7.18 "State Transition (when the sub clock is used)")" is deleted from 7.2.

•Page 101 of 604, description of the second paragraph in 7.2.1 is corrected as follows:
"When the main clock oscillator resumes running after an oscillator stop is detected, the PLL clock frequency may temporarily exceed the preset value before the PLL frequency synthesizer oscillation stabilizes. As soon as an oscillator stop is detected, the main clock oscillator should be stopped from resuming (set the CM05 bit in the CM0 register to 1) or the divide ratios of the base clock and peripheral clock source should be increased by a program. They can be set using bits BCD1 and BCD0 in the CCR register and bits PM36 and PM35 in the PM3 register."

•Page 105 of 604, description "f(XPLL)" in the third row of Figure 7.18 is corrected as follows:
"f(PLL)"

•Page 106 of 604, description "CM0 = 1" in the fourth row of Figure 7.19 is corrected as follows:
"CM05 = 1"

•Page 107 of 604, descriptions "CM31 = 1" in the first row and "CM10 = 0" for "Low speed mode" in the second row of Figure 7.20 are corrected as follows:
"CM31 = 0" and "CM10 = 1"

•Page 109 of 604, description of 7.7.2 is corrected as follows:
"The base clock stops in wait mode so that clocks generated by the base clock, the CPU clock and peripheral bus clock, stop running as well. Thus the CPU and watchdog timer, operated by these two clocks, also stop. Since the main clock, sub clock, PLL clock, and on-chip oscillator clock continue running, peripheral functions using these clocks also continue operating."

•Page 112 of 604, description in 7.7.3 is corrected as follows:
"In stop mode, all of the clocks, except for those that are protected, stop running. That is, the CPU and peripheral functions, operated by the CPU clock and peripheral clock, also stop. This mode saves the most power."

•Page 113 of 604, description of the first paragraph in 7.7.3.3 is modified as follows:
"The MCU exits stop mode by a hardware reset, NMI, low voltage detection interrupt, or a peripheral interrupt assigned to software interrupt number from 0 to 63."

•Page 121 of 604, description of Note 1 in Figure 10.1 is modified as follows:
"The peripheral interrupts are generated by the corresponding peripheral functions in the MCU."

•Page 122 of 604, descriptions in the second paragraph of (5) in 10.2 are corrected as follows:
"The stack pointer (SP) used for this interrupt differs depending on the software interrupt numbers. For software interrupt numbers 0 to 127, when an interrupt request is accepted, the U flag is saved and set to 0 to select the interrupt stack pointer (ISP) during the interrupt sequence. The saved data of the U flag is restored upon returning from the interrupt handler. For software interrupt numbers 128 to 255, the stack pointer does not change during the interrupt sequence."

•Page 124 of 604, description of 10.5 is corrected as follows:
"Each interrupt vector has a 4-byte memory space, in which the start address of the associated interrupt handler is stored. When an interrupt request is accepted, a jump to the address set in the interrupt vector takes place. Figure 10.2 shows an interrupt vector."

•Page 125 of 604, description in the Remarks for the BRK instruction in Table 10.1 is corrected as follows:
"If address FFFFFFE7h is FFh, a jump to the interrupt vector of software interrupt number 0 in the relocatable vector table takes place"

•Page 132 of 604, description for the IR bit below Figure 10.4 is corrected as follows:
"The IR bit becomes 1 (interrupt requested) when an interrupt request is generated; this bit setting is retained until the interrupt request is accepted. When the request is accepted and a jump to the corresponding interrupt vector takes place, the IR bit becomes 0 (no interrupt requested). The IR bit can be set to 0 by a program. This bit should not be set to 1."

•Page 136 of 604, description of Note 1 in Table 10.7 is corrected as follows:
"These are the values when the interrupt vectors are aligned to the addresses in multiples of 4 in the internal ROM. However, the condition does not apply to the fast interrupt."

•Page 139 of 604, Figure 10.8 is corrected as follows:

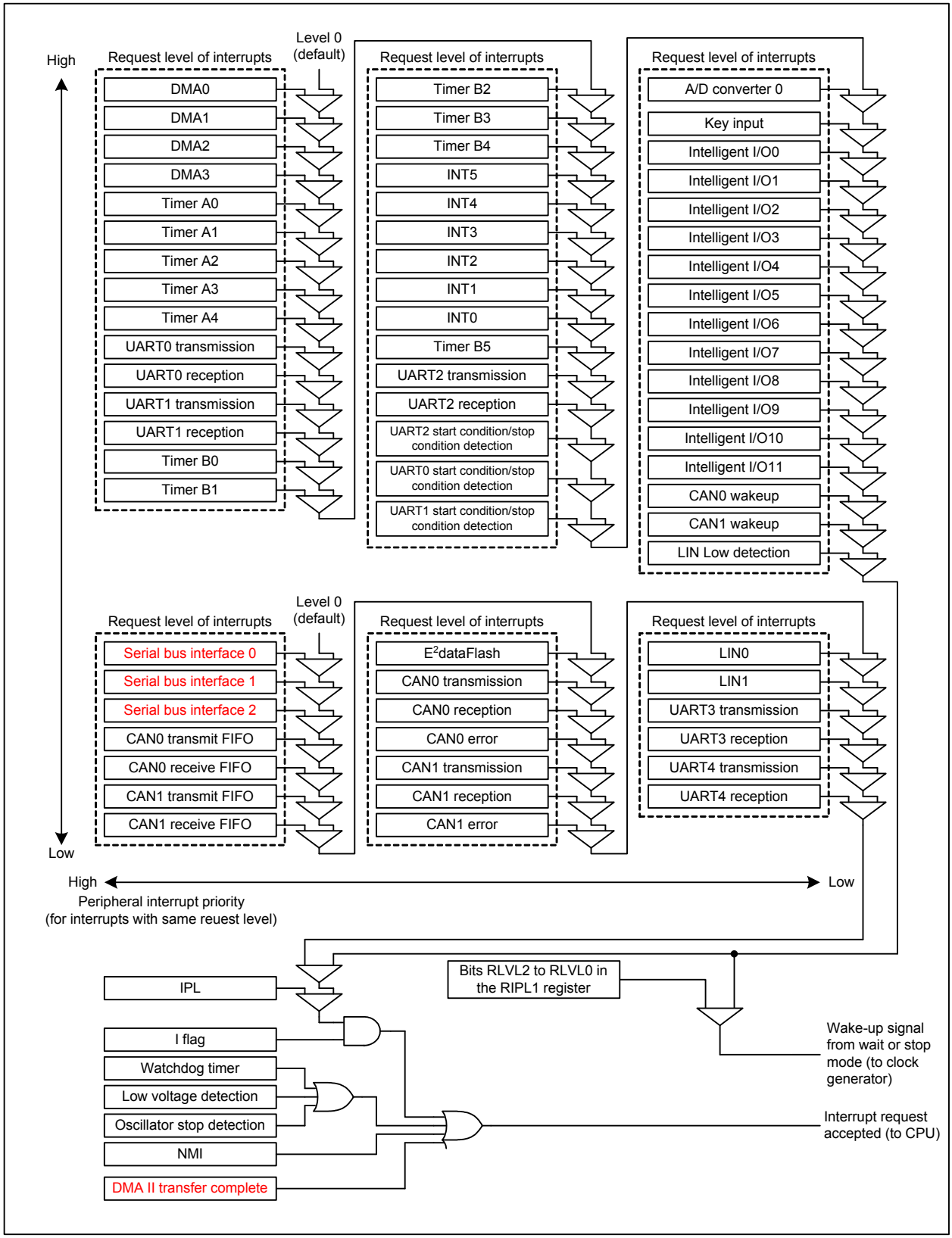


Figure 10.8 Priority Resolver

(Description “Bits RLVL2 to RLVL0 in the RIPL2 register” and associated signal lines are deleted from Figure 10.8)

•Page 143 of 604, register symbol “IIOiE” in line 16 of 10.13 is corrected as follows:
“IIOiIE”

•Page 144 of 604, descriptions for b0 and Note 3 in Figure 10.14 are corrected as follows:
b0: “No register bit; this bit is read as 1” (“should be written with 0 and” is deleted)
Note 3: “When this bit is function-assigned, it can only be set to 0. It should not be set to 1. To set it to 0, either the AND or BCLR instruction should be used; when the bit is not function-assigned (reserved), it should be set to 0.”

•Page 147 of 604, description of the second paragraph in 11. Watchdog Timer is corrected as follows:
“Select either an interrupt request or a reset with the CM06 bit in the CM0 register for when the watchdog timer underflows. Once the CM06 bit is set to 1 (reset), it cannot be changed to 0 (watchdog timer interrupt) by a program. It can be set to 0 only by a reset.”

•Page 147 of 604, register symbol “WKD” in line 9 of 11. Watchdog Timer is corrected as follows:
“WDK”

•Page 148 of 604, the following description is added to Figure 11.2 as Note 1:
“When the on-chip oscillator clock is used as the count source, the read value may be undefined due to a change in the count value while being read.”

•Page 150 of 604, the following description is added to Note 3 in Figure 11.5:
“The values set to these bits are reflected to registers WDK and PM2 when the WDTON bit is 0.”

•Page 152 of 604, expression “a value more than 00000001h” in the Specification of the DMA transfer startup in Table 12.1 is corrected as follows:
“a value other than 00000000h”

•Page 160 of 604, description of the first paragraph in 12.1 is corrected as follows:
“The transfer cycle is composed of bus cycles to read data from (source read) or to write data to (destination write) memory or an SFR.”

•Page 166 of 604, address “FFFFFFFh” in Note 1 of Table 13.1 is corrected as follows:
“FFFFFFFFh”

•Page 166 of 604, bit symbol “IIRLT” in the fifth bullet point of 13.1 is corrected as follows:
“IRLT”

•Page 168 of 604, expression “DMA II transfer complete interrupt vector address” in 13.1.2 and Figure 13.2 is corrected as follows:
“jump address for the DMA II transfer complete interrupt handler”

•Pages 168 and 171 of 604, expression “interrupt vector” in Figure 13.2 and 13.1.4 is corrected as follows:
“interrupt vector space”

•Page 169 of 604, description “jump address” in the seventh bullet point of 13.1.2 is corrected as follows:
“start address”

•Page 170 of 604, bit names of the OPER bit and bits CNT0 to CNT2 in Figure 13.3 are modified as follows:
OPER: “Calculation Result Transfer Select Bit”
CNT0 to CNT2: “Number of Transfers Setting Bit”

•Page 179 of 604, Figure 15.2 is corrected as follows:

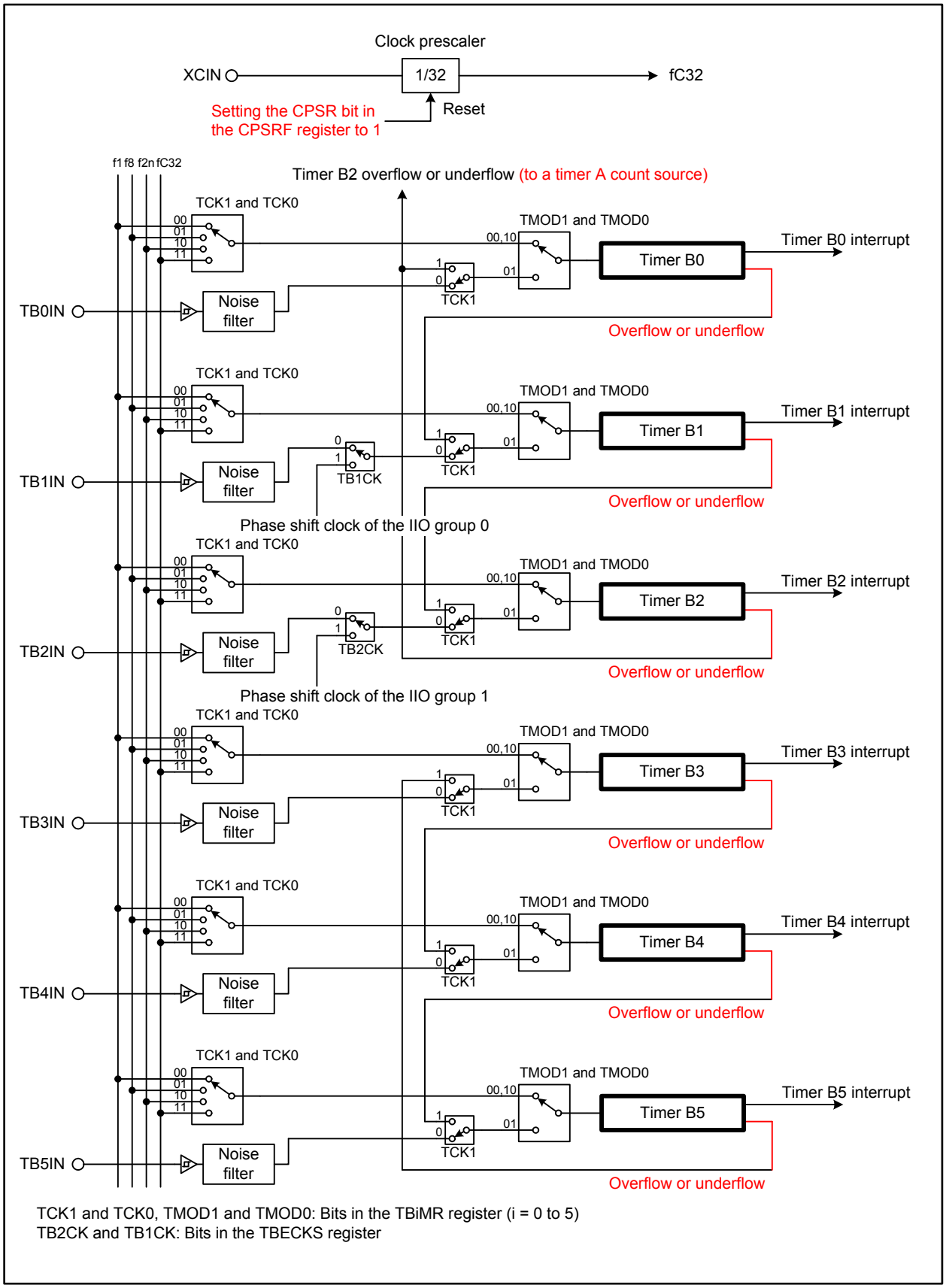


Figure 15.2 Timer B Configuration

•Page 183 of 604, expression “Counting” is deleted from bit names of bits TA0UD to TA4UD and the register name in Figure 15.7

•Page 191 of 604, bit name of the MR2 bit in Figure 15.12 is modified as follows:
“Increment/Decrement Switching Source Select Bit”

•Page 191 of 604, bit symbols “TAiTGH and TAiTGL” in Note 5 of Figure 15.12 are corrected as follows:
“TAjTGH and TAjTGL”

•Page 193 of 604, register symbol “TA4NR” in line 3 of 15.1.3 is corrected as follows:
“TA4MR”

•Page 194 of 604, descriptions of functions of the MR2 bit in Figure 15.15 are modified as follows:
“0: TAiOS bit in the ONSF register is enabled
1: Selected using bits TAiTGH and TAiTGL in the ONSF or TRGSR register”

•Page 196 of 604, descriptions of functions of the MR2 bit in Figure 15.16 are modified as follows:
“0: TAiS bit in the ONSF register is enabled
1: Selected using bits TAiTGH and TAiTGL in the ONSF or TRGSR register”

•Page 211 of 604, register symbol “TBjMR” in the first bullet point of 15.3.3.2 is corrected as follows:
“TABSR or TBSR”

•Page 211 of 604, expression “TBj interrupt handler” in the eighth bullet point of 15.3.3.2 is changed as follows:
“timer Bj interrupt handler”

•Page 215 of 604, descriptions of functions of the INV13 bit in Figure 16.3 are corrected as follows:
“0: Timer A1 reload control signal is 0
1: Timer A1 reload control signal is 1”

•Page 215 of 604, description of Note 1 in Figure 16.3 is corrected as follows:
“Set this register after setting the PRC1 bit in the PRCR register to 1 (write enabled). Also, rewrite this register while timers A1, A2, A4, and B2 are stopped.”

•Page 220 of 604, descriptions of functions of bits MR2 and MR3 in Figure 16.8 are corrected as follows:
MR2: “No register bit; should be written with 0 and read as undefined value”
MR3: “Disabled when using the three-phase motor control timers. Should be written with 0 and read as undefined value”

•Page 221 of 604, description of function of the PWCON bit in Figure 16.9 is corrected as follows:
“1: The underflow of timer B2 when the reload control signal for timer A1 is 0”

•Page 222 of 604, description “The sum of setting values for registers TAi and TAi1 should be identical to the setting value of the TB2 register in this mode.” is deleted from lines 8 to 9 of 16.3

•Page 222 of 604, description in line 11 of 16.3 is corrected as follows:
“Figure 16.11 shows registers TA1M, TA2M, TA4M, TA11M, TAM21M, and TA41M in this function.”

•Page 227 of 604, bit symbol “INV06” in Note 3 of Figure 16.16 is corrected as follows:
“INV16”

•Page 228 of 604, register symbol “INV1” in Note 2 of Figure 16.18 is corrected as follows:
“INVC1”

•Page 230 of 604, description of 16.6.1 is corrected as follows:
"When a low signal is applied to the $\overline{\text{NMI}}$ pin with the following bit settings, pins TA1OUT, TA2OUT, and TA4OUT become high-impedance: the PM24 bit in the PM2 register is 1 (NMI enabled), the SDE bit in the IOBC register is 1 (shutdown enabled), the INV02 bit in the INVC0 register is 1 (three-phase motor control timers used), and the INV03 bit is 1 (three-phase motor control timer output enabled)."

•Page 230 of 604, description of 16.6.2 is corrected as follows:
"Do not write to the TAi1 register (i = 1, 2, 4) before and after timer B2 underflows. Before writing to the TAi1 register, read the TB2 register to verify that sufficient time remains until timer B2 underflows. Then, immediately write to the TAi1 register so no interrupt handling is performed during this write procedure. If the TB2 register indicates little time remains until the underflow, write to the TAi1 register after timer B2 underflows."

•Page 236 of 604, descriptions for the CRD bit in Figure 17.5 are modified as follows:

Bit Symbol	Bit Name	Function	RW
CRD	CTS Function Disable Bit	0: CTS function enabled 1: CTS function disabled	RW

•Page 236 of 604, Note 1 "Bits CNT3 to CNT0 in the TCSPR register select a divide ratio from two options: no division (n = 0) or divide-by-2n (n = 1 to 15)." is deleted from Figure 17.5.

•Page 238 of 604, description of function of the UiIRS bit in Figure 17.7 is modified as follows:
"0: Transmit buffer is empty (TI = 1)
1: Transmission is completed (TXEPT = 1)"

•Page 240 of 604, description of function of the SWC bit in Figure 17.11 is modified as follows:
"0: No wait-state/wait-state cleared
1: Hold the SCLi pin low after the eighth bit is received"

•Page 241 of 604, description "UiBRG count source" in the function of bits DL0 to DL2 in Figure 17.12 is corrected as follows:
"baud rate generator count source"

•Page 242 of 604, description of function of the SWC9 bit in Figure 17.13 is modified as follows:
"0: No wait-state/wait-state cleared
1: Hold the SCLi pin low after the ninth bit is received"

•Pages 242 and 268 of 604, bit symbol "STARREQ" in Note 3 of Figure 17.13 and line 1 of 17.3.2 is corrected as follows:
"STAREQ"

•Page 248 of 604, description "TXEPT flag" in Figure 17.18 is corrected as follows:
"TXEPT bit"

•Page 248 of 604, bit symbol "UiRS" in the fourth dash of Figure 17.18 is corrected as follows:
"UiIRS"

•Pages 256 and 257 of 604, descriptions of functions of the UiIRS bit in Figures 17.23 and 17.24 are corrected as follows:
0: "(an interrupt request is generated when the transmit buffer is empty)"
1: "(an interrupt request is generated when transmission is completed)"

•Pages 260 and 261 of 604, description “Transmit/receive clock” in Figures 17.27 and 17.28 is corrected as follows:
“CLKi”

•Page 278 of 604, description of the fourth dash in 17.5.3.1 is replaced as follows:
“- The TE bit in the UiC1 register is 1 (transmission enabled).
- The RE bit in the UiC1 register is 1 (reception enabled). This bit setting is not required in transmit operation only.
- The TI bit in the UiC1 register is 0 (data held in the UiTB register).”

•Page 292 of 604, description in 18.1.5 is modified as follows:
“In repeat sweep mode 1, the analog voltage applied to eight selected pins including one to four prioritized pins is repeatedly converted into a digital code. Table 18.6 lists specifications of repeat sweep mode 1.”

•Page 292 of 604, description for the function in Table 18.6 is modified as follows:
“The analog voltage applied to eight selected pins including one to four prioritized pins is repeatedly converted into a digital code. The prioritized pins are selected by setting bits SCAN1 and SCAN0 in the AD0CON1 register and bits APS1 and APS0 in the AD0CON2 register”

•Page 301 of 604, description “AD0i register” in the ninth bullet point of 18.3.2 is modified as follows:
“AD00 register”

•Page 304 of 604, description “CRC_CCITT” in line 2 of 20. CRC Calculator is corrected as follows:
“CRC-CCITT”

•Page 304 of 604, Figure 20.1 is corrected as follows:

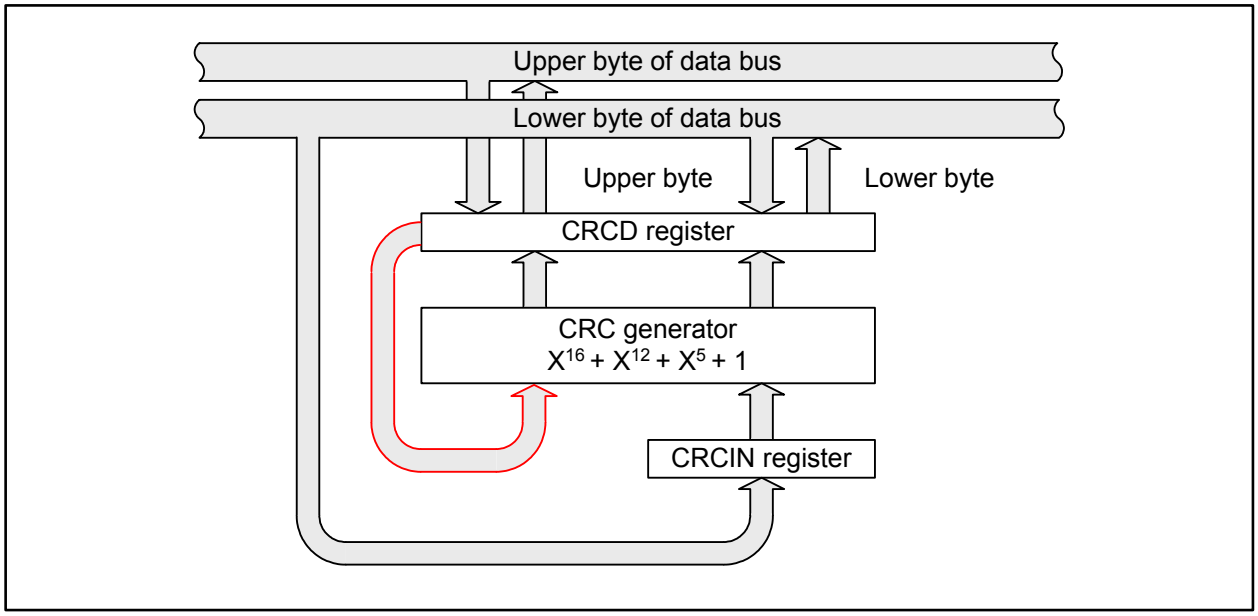


Figure 20.1 CRC Calculator Block Diagram

•Pages 312 and 313 of 604, descriptions “Request from the $\overline{\text{INT0}}$ pin” in Figure 22.1 and “Request from the $\overline{\text{INT1}}$ pin” in Figure 22.2 are corrected as follows:
Figure 22.1: “Request from the $\overline{\text{INT0}}$ pin or the $\overline{\text{INT1}}$ pin”
Figure 22.2: “Request from the $\overline{\text{INT0}}$ pin or the $\overline{\text{INT1}}$ pin”

•Page 316 of 604, descriptions for bits RST2, UD0, and UD1 in Figure 22.5 are modified as follows:

Bit Symbol	Bit Name	Function	RW
RST2	Base Timer Reset Source Select Bit 2	0: No reset 1: Low signal input into the $\overline{INT0}$/$\overline{INT1}$ pin (3)	RW

UD0	Increment/Decrement Control Bit	b6 b5 0 0 : Increment mode 0 1 : Increment/decrement mode 1 0 : Two-phase pulse signal processing mode (4) 1 1 : Do not use this combination	RW
UD1			RW

•Page 316 of 604, description of Note 3 in Figure 22.5 is modified as follows:
"The base timer is reset by an input of low signal to the external interrupt input pin selected for the UDiZ signal by the IFS2 register."

•Page 317 of 604, Note 3 "The GOC bit becomes 0 after gating is cleared." is deleted from Figure 22.6.

•Page 323 of 604, descriptions in the second bullet point for the reset conditions in Table 22.2 are corrected as follows:
"An input of low signal into the external interrupt pin ($\overline{INT0}$ or $\overline{INT1}$) as follows:
for group 0: selected using bits IFS23 and IFS22 in the IFS2 register
for group 1: selected using bits IFS27 and IFS26 in the IFS2 register"

•Page 323 of 604, description in the first bullet point for the selectable functions in Table 22.2 is corrected as follows:
"The base timer starts counting when the BTS bit is set to 1. When the base timer reaches FFFFh, it starts decrementing. When the RST1 bit in the GiBCR1 register is 1 (the base timer is reset by matching with the GiPO0 register), the timer counter starts decrementing two counts after the base timer value matches the GiPO0 register setting. When the timer counter reaches 0000h, it starts incrementing again (refer to Figure 22.17)."

•Page 324 of 604, description "Low signal input to the $\overline{INTi}$ pin" in Figure 22.15 is corrected as follows:
"Low signal input to the $\overline{INT0}$/$\overline{INT1}$ pin"

•Pages 334, 336, and 339 of 604, description "Input to the IIOi_j pin" in Figures 22.22 to 22.24 is corrected as follows:
"IIOi_j pin"

•Page 345 of 604, bit symbol "SBUMS" in 23. Serial Bus Interface is corrected as follows:
"SSUMS"

•Pages 346 and 348 of 604, pin name "SSiCK" and register symbol "SSiRDR" in Tables 23.1 and 23.2 are corrected as follows:
"SSCKi" and "SSiTDR"

•Page 348 of 604, descriptions "SSIi (I): Data input pin" and "SSOi (O): Data output pin" for the I/O pins in Table 23.2 are corrected as follows:
"SSIi (I/O): Data I/O pin" and "SSOi (I/O): Data I/O pin"

•Page 350 of 604, description “b6 b5 b4” in Figure 23.3 is corrected as follows:
“b2 b1 b0”

•Page 380 of 604, descriptions “Break dominant” and “Break delimiter” in Table 24.1 are modified as follows:
“Transmit break length” and “Transmit break delimiter length”

•Page 384 of 604, the following description is added to Figure 24.3 as Note 1:
“No new interrupt is generated by the input signal low detection when any one of these bits is 1.”

•Page 386 of 604, description in Note 4 of Figure 24.7 is modified as follows:
“The LD bit in the LST register becomes 1 and an interrupt request is generated in the following cases:
- When the falling edge of the input signal is detected when this bit is 1.
- When this bit is set to 1 while the input signal is low.”

•Page 389 of 604, bit names “Break Transmission Setting Bit” and “Break Delimiter Transmission Setting Bit” in Figure 24.10 are modified as follows:
Bits BLT0 to BLT3: “Transmit Break (Low) Length Setting Bit”
Bits BDT0 and BDT1: “Transmit Break Delimiter (High) Length Setting Bit”

•Pages 393 and 394 of 604, description in Note 1 of Figures 24.16 and 24.17 is modified as follows:
“These bits do not become 0 automatically. Set them to 0 by a program. Writing 1 to these bits has no effect.”

•Page 393 of 604, the following description is added to Figure 24.16 as Note 3:
“When this bit is 1, no new interrupt request is generated in the following cases:
- When the LD bits of other channels become 1.
- When the conditions for the LD bit to become 1 are met in its channel.”

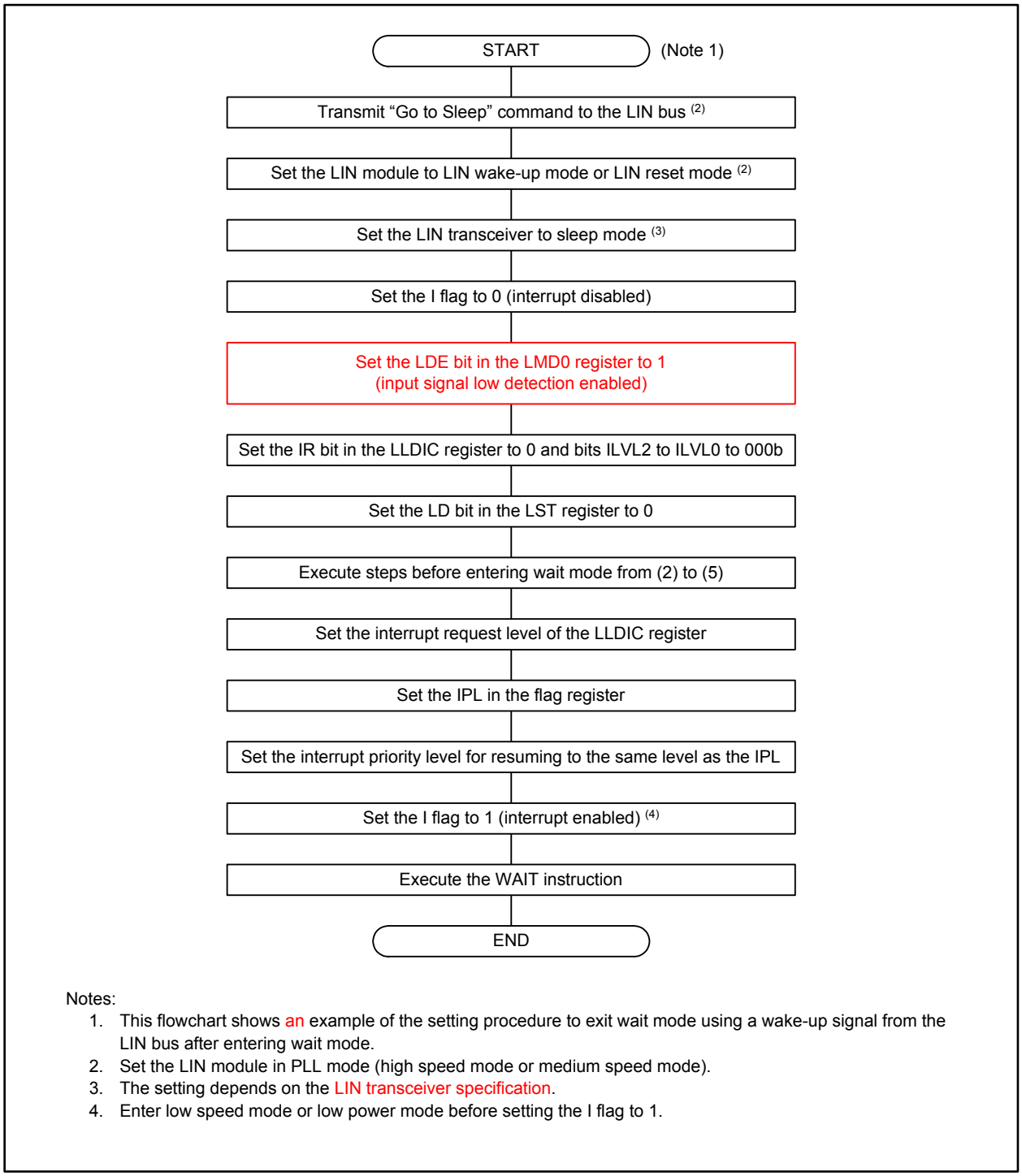
•Page 398 of 604, bit symbols “BFTL3 to BFTL0” and “BFTD1 and BFTD0” in Table 24.3 are corrected as follows:
“BLT3 to BLT0” and “BDT1 to BDT0”

•Page 399 of 604, description in (4) for “LIN Module Processing” in Table 24.4 is changed as follows:
“Transmit Data 2, then the next interbyte space
Transmit data 3, then the next interbyte space
(Repeat this process for the data length specified in bits RFDL3 to RFDL0 in the LRFC register. Go to (6) if an error occurs.)”

•Page 400 of 604, description in (4) for “LIN Module Processing” in Table 24.5 is changed as follows:
“Receive Data 2 due to start bit detection
Receive Data 3 due to start bit detection
(Repeat this process for the data length specified in bits RFDL3 to RFDL0 in the LRFC register. Abort the reception and go to (5) if an error occurs. Checksum judgement is not performed in this case.)”

•Page 402 of 604, description “BPR0” in Note 1 of Table 24.6 is corrected as follows:
“BRP0”

•Page 409 of 604, Figure 24.30 is corrected as follows:



Notes:
1. This flowchart shows an example of the setting procedure to exit wait mode using a wake-up signal from the LIN bus after entering wait mode.
2. Set the LIN module in PLL mode (high speed mode or medium speed mode).
3. The setting depends on the LIN transceiver specification.
4. Enter low speed mode or low power mode before setting the I flag to 1.

Figure 24.30 Example of Setting Before Transition to Wait Mode

•Page 410 of 604, description for the Input signal Low detection row in Table 24.8 is modified as follows:
“When the falling edge of input signal at the LINjIN pin is detected with the setting of the LDE bit in the LMD0 register to 1 (input signal low detection enabled), or when setting the LDE bit to 1 while the LINjIN pin is low”

•Page 413 of 604, the following description is added to line 9 of 24.11:
“No new interrupt request is generated by the other sources if any of them is 1 since multiple interrupt sources are aggregated.”

•Page 413 of 604, Figure 24.32 is corrected as follows:

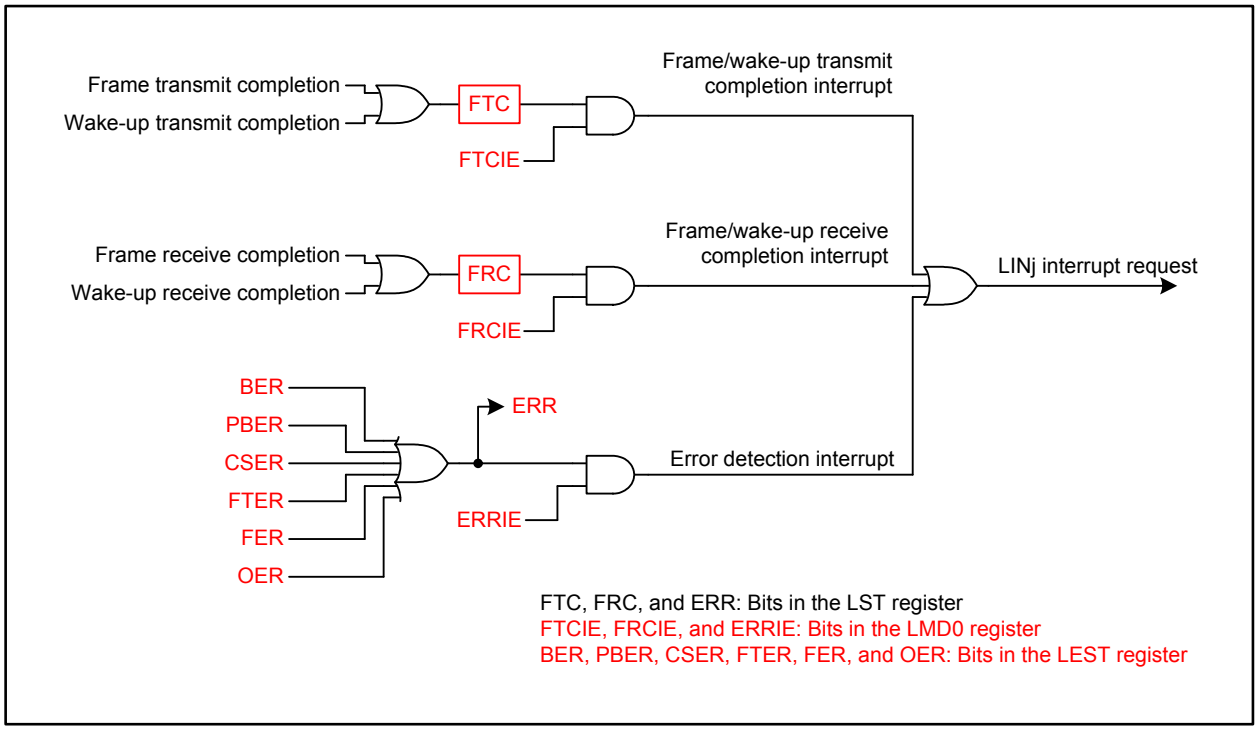


Figure 24.32 LINj Interrupt Block Diagram (j = 0, 1)

•Page 413 of 604, Figure 24.33 is corrected as follows:

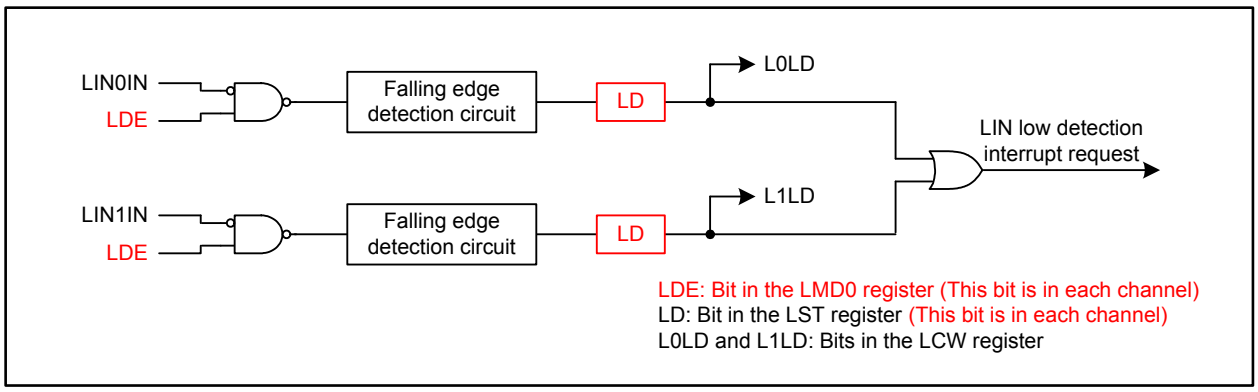


Figure 24.33 LIN Low Detection Interrupt Block Diagram

•Page 414 of 604, table number “Table 25.1” in 25. CAN Module is corrected as follows:
“Tables 25.1 and 25.2”

•Page 418 of 604, descriptions for the RBOC bit in Figure 25.2 are modified as follows:

Bit Symbol	Bit Name	Function	RW
RBOC	Forced Recovery From Bus-off Bit (4)	0: Nothing occurred 1: Forced recovery from bus-off (5)	RW

•Page 437 of 604, description of Note 2 in Figure 25.11 is modified as follows:
“When setting the RFE bit to 0, set the RFMLF bit to 0 as well.”

•Page 444 of 604, description of function of b7 in Figure 25.17 is corrected as follows:
b7: “No register bit; this bit is read as 0” (“should be written with 0 and” is deleted)

•Page 468 of 604, description of the first paragraph in 25.2.3 is modified as follows:
“CAN sleep mode is used for reducing current consumption by stopping the clock supply to the CAN module. After a MCU reset, the CAN module starts from CAN sleep mode.”

•Page 469 of 604, register symbol “CiSTR” in 25.2.4 is corrected as follows:
“CiTCR”

•Page 471 of 604, q value “q = 1, 2, 3, 4” in Figure 25.36 is corrected as follows:
“q = 2, 3, 4”

•Page 483 of 604, description in the first paragraph of 26. I/O Pins is corrected as follows (refer to TN_16C_A198A/E):
“Each pin of the MCU functions as a programmable I/O port or an I/O pin for internal peripheral functions. These functions can be switched by the function select registers. The pull-up resistors are enabled for every group of four pins. However, a pull-up resistor is separated from other peripheral functions even if it is enabled, when a pin functions as an output pin.” (“or an analog I/O pin” is deleted)

•Page 483 of 604, Figure 26.1 is corrected as follows (refer to TN_16C_A198A/E):

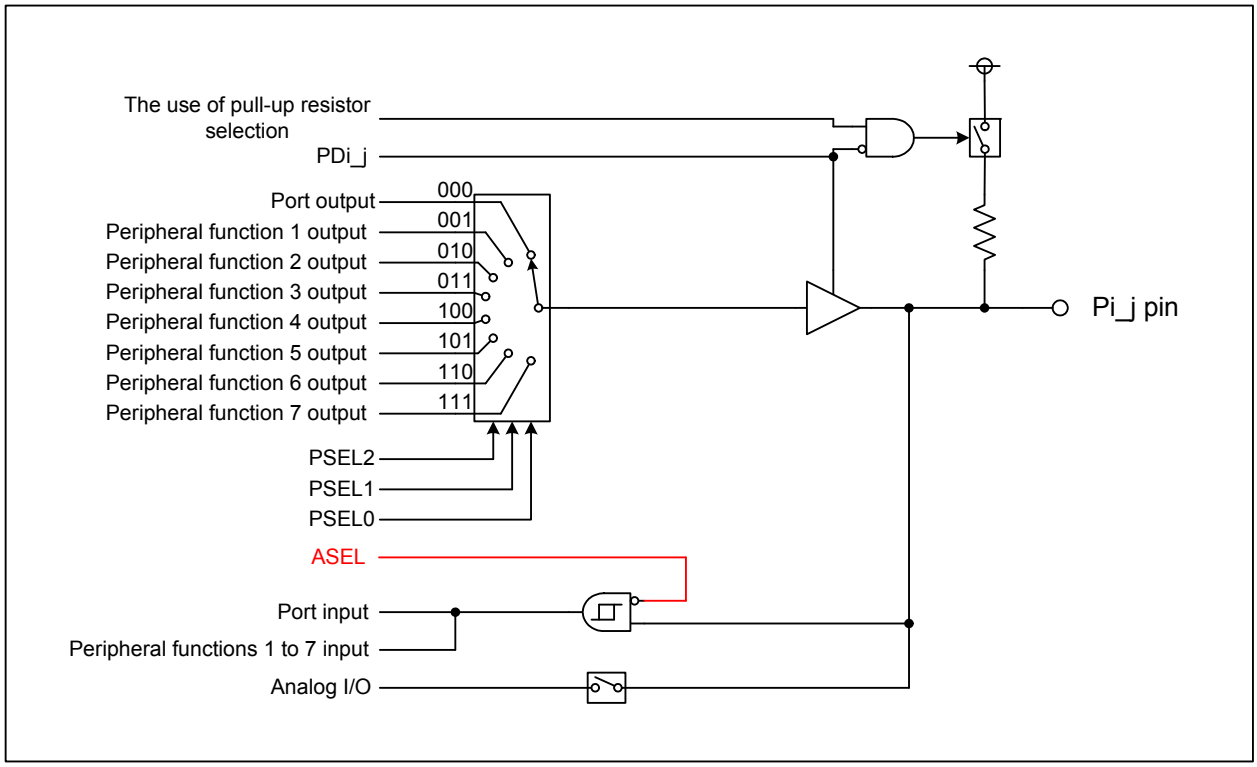


Figure 26.1 Typical I/O Pin Block Diagram (i = 0 to 10; j = 0 to 7)

•Page 483 of 604, description in the last paragraph of 26. I/O Pins is corrected as follows:
“The input-only port P8_5, which shares a pin with $\overline{\text{NMI}}$ has neither the function select register nor bit 5 in the PD8 register. Port P9_1 also functions as an input-only port. The function select register and bit 1 in the PD9 register are reserved. Port P9 is protected from unexpected write accesses by the PRC2 bit in the PRCR register. Ports P3, P7, and P8 are protected from unexpected write accesses by the PRC30 bit in the PRCR3 register (refer to 9. “Protection”).”

•Pages 487, 490, and 491 of 604, descriptions “IIO0 output” and “IIO1 output” in Figures 26.4, 26.7, and 26.8 are changed as follows:
“IIO0_i output” and “IIO1_i output”

•Page 489 of 604, description “PD3_i register” in line 4 below Figure 26.6 is corrected as follows:
“PD3_i bit”

•Page 509 of 604, descriptions in Table 27.3 are corrected as follows:

Protection Type	Lock Bit Protection	ROM Code Protection	ID Code Protection
Protected operations	Erase, write	Read, write	Read, erase, write

(“erase” is deleted from the ROM Code Protection column)

Protection deactivated by	Setting the LBD bit in the FMR register to 1 (lock bit protection disabled), or by erasing the blocks whose lock bits are set to 0 to permanently deactivate the protection	Erasing all blocks whose protect bits are set to 0	Inputting a proper ID code to the serial programmer

(“by using the serial programmer” is deleted from the ROM Code Protection column)

•Page 509 of 604, description “use the serial programmer to” is deleted from line 3 of 27.2.2.

•Page 510 of 604, Figure 27.2 is corrected as follows:

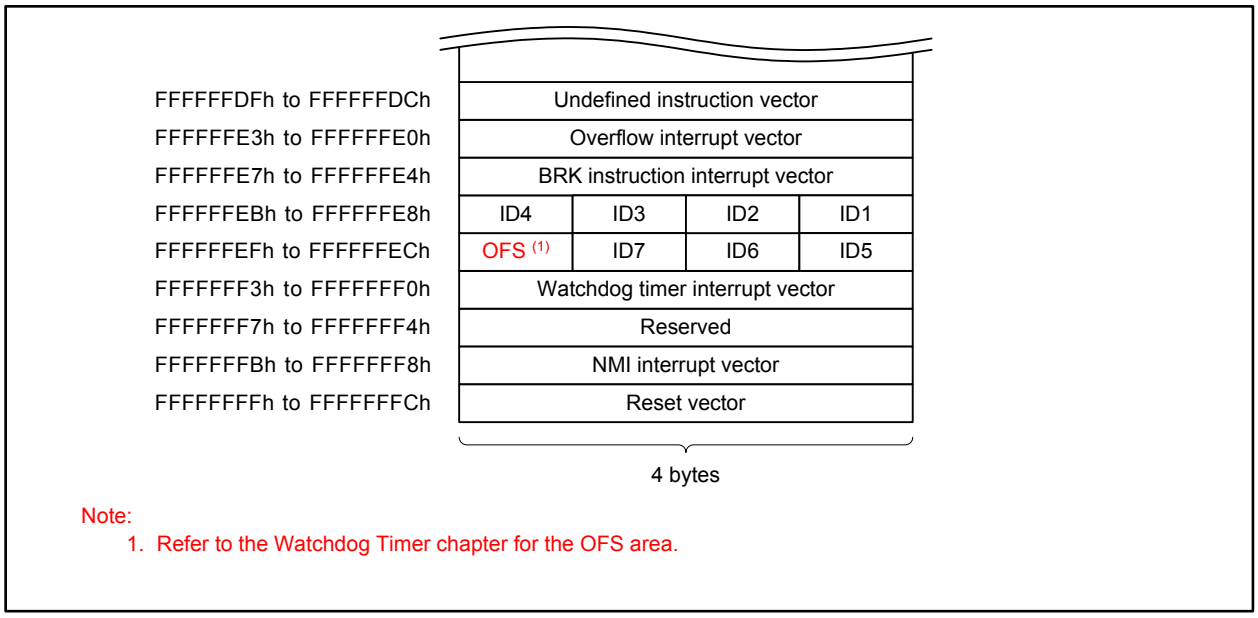


Figure 27.2 Addresses for ID Code Stored

•Page 512 of 604, descriptions in Table 27.5 are modified as follows:

Restrictions on software commands	None	• Do not execute either the program command or the block erase command for blocks where the rewrite control programs are written to • Do not execute the enter read status register mode command • Execute the enter read lock bit status mode command in RAM • Execute the enter read protect bit status mode command in RAM

Flash memory state detection by	• Reading the FMSR0 register by a program • Executing the enter read status register mode command to read data	• Reading the FMSR0 register by a program

•Pages 518 and 520 of 604, descriptions in Figures 27.12 and 27.13 are corrected as follows:

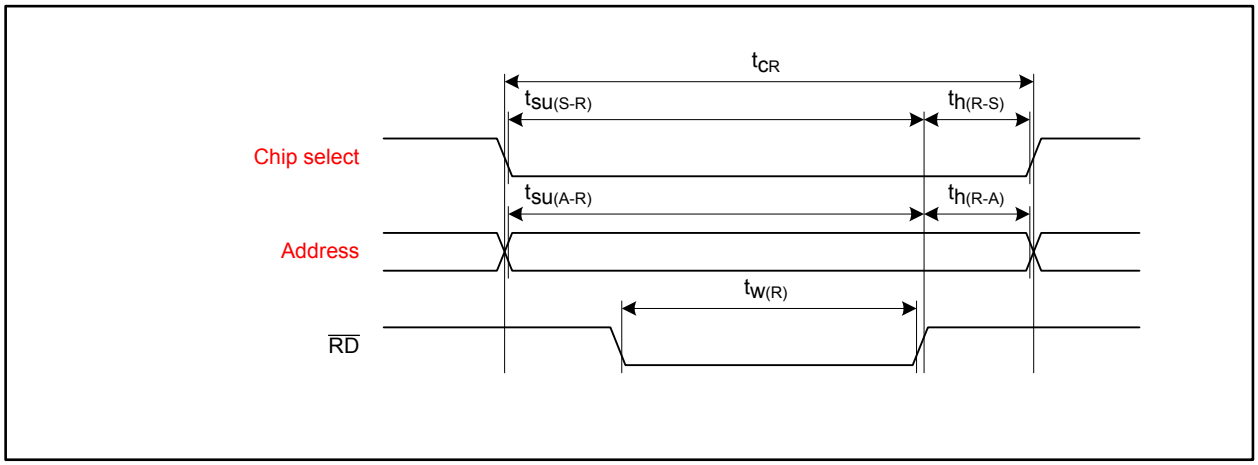


Figure 27.12 Read Timing

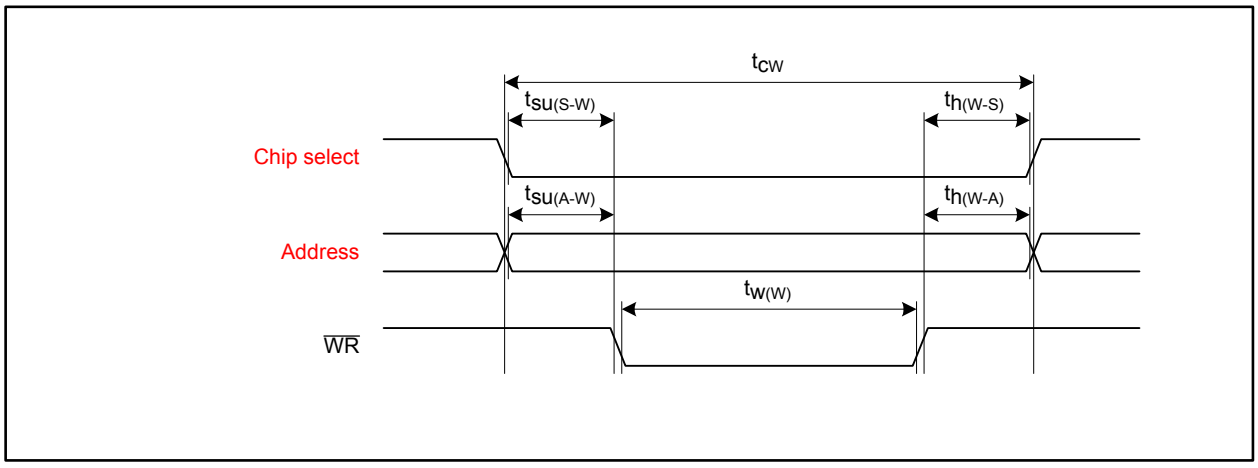


Figure 27.13 Write Timing

•Page 538 of 604, the following description is added to Note 3 of Figure 28.2:
"However, the registers are not initialized."

•Page 538 of 604, description "This mode setting prevents data from being overwritten if a program goes out of control." is deleted from Note 5 in Figure 28.2.

•Page 544 of 604, description "EERR bit in the E2FS0 register is 1?" in Figure 28.12 is corrected as follows:
"EERR bit in the E2FS0 register is 0?"

•Pages 553 and 554 of 604, description "Programming and erasure endurance of flash memory" in Tables 29.8 and 29.9 is changed as follows:
"Program and erase cycles"

•Pages 553 and 554 of 604, unit "times" for "Programming and erasure endurance of flash memory" in Tables 29.8 and 29.9 and Note 1 is corrected as follows:
"Cycles"

•Page 557 of 604, descriptions in Figure 29.5 are corrected as follows:

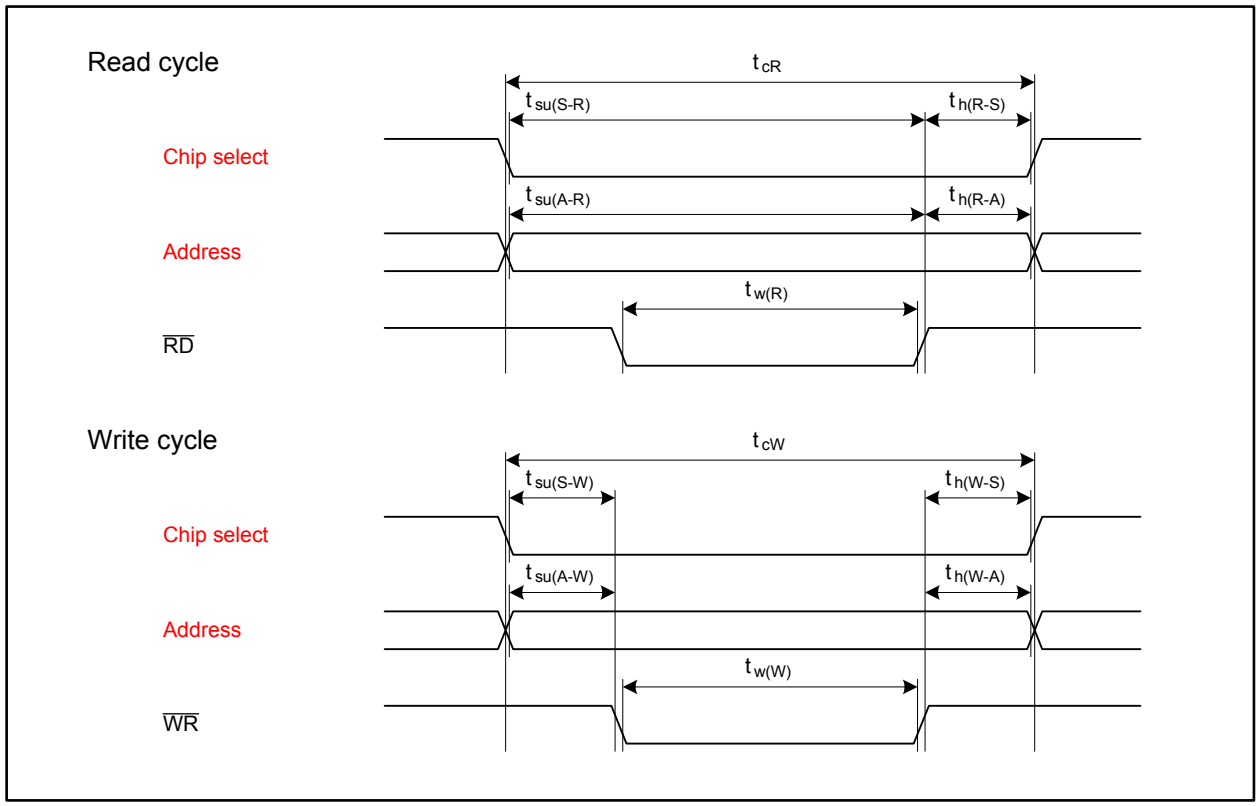


Figure 29.5 Flash Memory CPU Rewrite Mode Timing

•Page 591 of 604, register symbol "TBjMR" in the first bullet point of 30.6.3.2 is corrected as follows:
"TABSR or TBSR"

•Page 591 of 604, expression "TBj interrupt handler" in the eighth bullet point of 30.6.3.2 is changed as follows:
"timer Bj interrupt handler"

•Page 592 of 604, description of 30.7.1 is corrected as follows:
"When a low signal is applied to the $\overline{\text{NMI}}$ pin with the following bit settings, pins TA1OUT, TA2OUT, and TA4OUT become high-impedance: the PM24 bit in the PM2 register is 1 (NMI enabled), the SDE bit in the IOBC register is 1 (shutdown enabled), the INV02 bit in the INVC0 register is 1 (three-phase motor control timers used), and the INV03 bit is 1 (three-phase motor control timer output enabled)."

•Page 592 of 604, descriptions in 30.7.2 are corrected as follows:
"Do not write to the TAi1 register (i = 1, 2, 4) before and after timer B2 underflows. Before writing to the TAi1 register, read the TB2 register to verify that sufficient time remains until timer B2 underflows. Then, immediately write to the TAi1 register so no interrupt handling is performed during this write procedure. If the TB2 register indicates little time remains until the underflow, write to the TAi1 register after timer B2 underflows."

•Page 593 of 604, description of the fourth dash in 30.8.3.1 is replaced as follows:
"- The TE bit in the UiC1 register is 1 (transmission enabled).
- The RE bit in the UiC1 register is 1 (reception enabled). This bit setting is not required in transmit operation only.
- The TI bit in the UiC1 register is 0 (data held in the UiTB register)."

•Page 595 of 604, description "AD0i register" in the ninth bullet point of 30.9.2 is modified as follows:
"AD00 register"