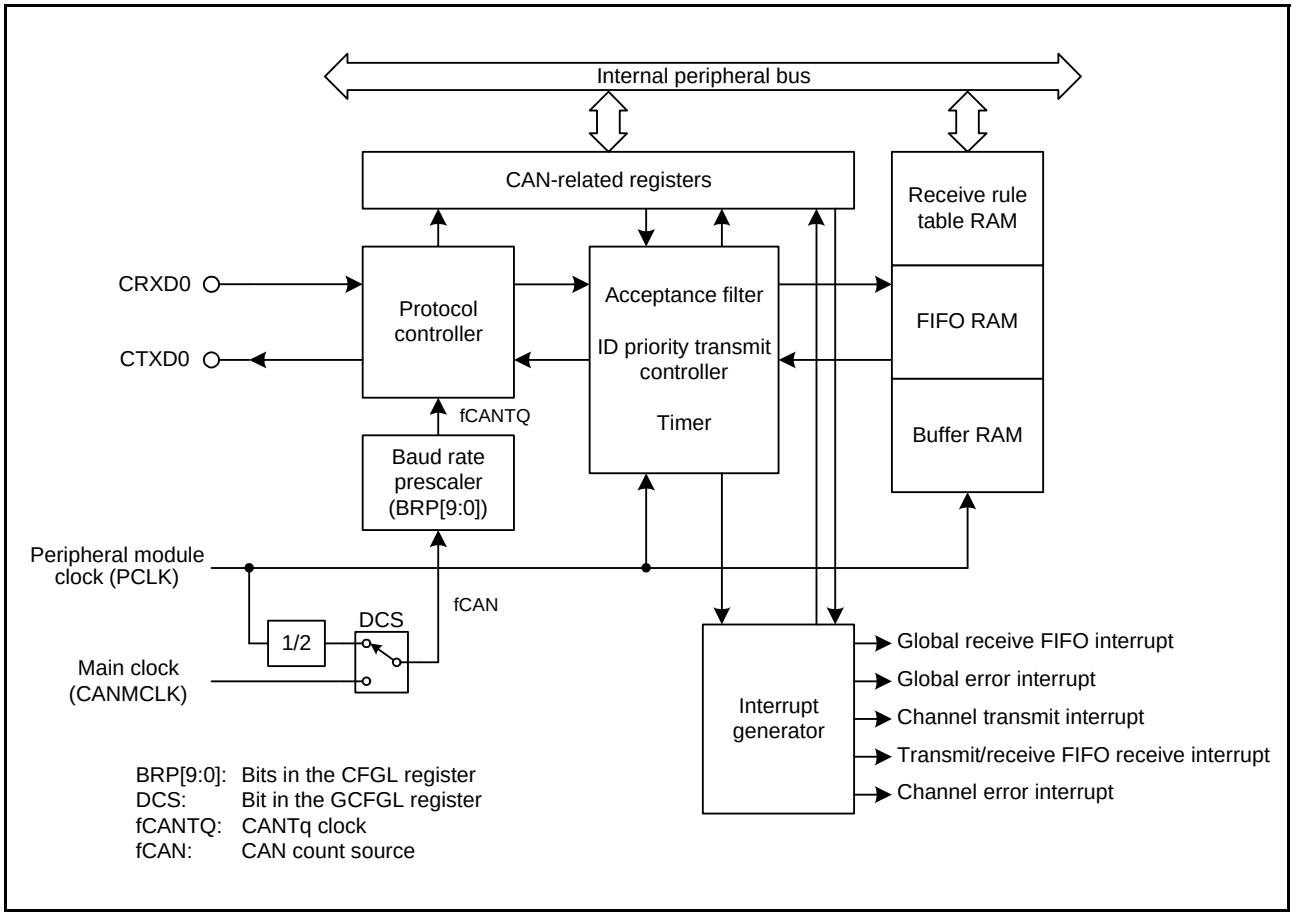



**Figure 36.1 CAN Module Block Diagram**

- CRXD0/CTXD0: CAN input/output pins
- Protocol controller: Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, and error handling, etc.
- Receive rule table RAM: Stores the rules for filtering received messages. Each receive rule specifies an ID/ frame format/data length code of the message to be received, a label to be attached to the message that has passed through the filter, and the location of such message to be stored.
- FIFO RAM: Includes three 16-stage FIFO buffers. There are two FIFOs for reception only and one FIFO for transmission or reception.
- Buffer RAM: Used as a transmit and receive buffer. There are 4 buffers for transmission and 16 buffers for reception.
- Acceptance filter: Performs filtering of received messages.
- Timer: There are a timer for timestamp function during reception and a timer which determines the message transmission intervals while using the transmit FIFO buffer.

**Table 36.2 I/O Pins of the CAN Module**

| Pin Name | I/O | Description |
|---|---|---|
| CRXD0 | Input | Receive data input pins of the RSCAN0 |
| CTXD0 | Output | Transmit data output pins of the RSCAN0 |

## 36.2 Register Descriptions

### 36.2.1 Bit Configuration Register L (CFGL)

Address(es): RSCAN0.CFGL 000A 8300h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | BRP[9:0] | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b9 to b0 | BRP[9:0] | Prescaler Division Ratio Set | When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1. | R/W |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Modify the CFGL register only in channel reset mode or channel halt mode. Set this register in channel reset mode before making a transition to channel communication mode or channel halt mode. For setting bit timing, see section 36.9, Initial Settings.

**BRP[9:0] Bits (Prescaler Division Ratio Set)**

The CAN Tq clock (fCANTQ) is obtained by the CAN clock source (fCAN) and setting the clock division ratio with the BRP[9:0] bits and one clock cycle of the CAN Tq clock is 1 Time Quantum (Tq).

## 36.2.2 Bit Configuration Register H (CFGH)

Address(es): RSCAN0.CFGH 000A 8302h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | SJW[1:0] | | — | TSEG2[2:0] | | | TSEG1[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | TSEG1[3:0] | Time Segment 1 Control | b3 b0<br>0 0 0 0: Setting prohibited<br>0 0 0 1: Setting prohibited<br>0 0 1 0: Setting prohibited<br>0 0 1 1: 4 Tq<br>0 1 0 0: 5 Tq<br>0 1 0 1: 6 Tq<br>0 1 1 0: 7 Tq<br>0 1 1 1: 8 Tq<br>1 0 0 0: 9 Tq<br>1 0 0 1: 10 Tq<br>1 0 1 0: 11 Tq<br>1 0 1 1: 12 Tq<br>1 1 0 0: 13 Tq<br>1 1 0 1: 14 Tq<br>1 1 1 0: 15 Tq<br>1 1 1 1: 16 Tq | R/W |
| b6 to b4 | TSEG2[2:0] | Time Segment 2 Control | b6 b4<br>0 0 0: Setting prohibited<br>0 0 1: 2 Tq<br>0 1 0: 3 Tq<br>0 1 1: 4 Tq<br>1 0 0: 5 Tq<br>1 0 1: 6 Tq<br>1 1 0: 7 Tq<br>1 1 1: 8 Tq | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b9, b8 | SJW[1:0] | Resynchronization Jump Width Control | b9 b8<br>0 0: 1 Tq<br>0 1: 2 Tq<br>1 0: 3 Tq<br>1 1: 4 Tq | R/W |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Modify the CFGH register only in channel reset mode or channel halt mode. Set this register in channel reset mode before making a transition to channel communication mode or channel halt mode. For setting bit timing, see section 36.9, Initial Settings.

### TSEG1[3:0] Bits (Time Segment 1 Control)

These bits are used to specify a Tq value for the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1). A value of 4 Tq to 16 Tq can be set.

### TSEG2[2:0] Bits (Time Segment 2 Control)

These bits are used to specify a Tq value for the length of phase buffer segment 2 (PHASE_SEG2).
A value of 2 Tq to 8 Tq can be set. Set a value smaller than the value of the TSEG1[3:0] bits.

**SJW[1:0] Bits (Resynchronization Jump Width Control)**

These bits are used to specify a Tq value for the resynchronization jump width. A value of 1 Tq to 4 Tq can be set. Set a value equal to or smaller than the value of the TSEG2[3:0] bits.

## 36.2.3 Control Register L (CTRL)

Address(es): RSCAN0.CTRL 000A 8304h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ALIE | BLIE | OLIE | BORIE | BOEIE | EPIE | EWIE | BEIE | — | — | — | — | RTBO | CSLPR | CHMDC[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CHMDC[1:0] | Mode Select | b1 b0<br>0 0: Channel communication mode.<br>0 1: Channel reset mode.<br>1 0: Channel halt mode.<br>1 1: Setting prohibited. | R/W |
| b2 | CSLPR | Channel Stop Mode | 0: Other than channel stop mode.<br>1: Channel stop mode. | R/W |
| b3 | RTBO | Forcible Return from Bus-off | When this bit is set to 1, forcible return from the bus off state is made. This bit is read as 0. | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | BEIE | Protocol Error Interrupt Enable | 0: Protocol error interrupt is disabled.<br>1: Protocol error interrupt is enabled. | R/W |
| b9 | EWIE | Error Warning Interrupt Enable | 0: Error warning interrupt is disabled.<br>1: Error warning interrupt is enabled. | R/W |
| b10 | EPIE | Error Passive Interrupt Enable | 0: Error passive interrupt is disabled.<br>1: Error passive interrupt is enabled. | R/W |
| b11 | BOEIE | Bus Off Entry Interrupt Enable | 0: Bus off entry interrupt is disabled.<br>1: Bus off entry interrupt is enabled. | R/W |
| b12 | BORIE | Bus Off Recovery Interrupt Enable | 0: Bus off recovery interrupt is disabled.<br>1: Bus off recovery interrupt is enabled. | R/W |
| b13 | OLIE | Overload Frame Transmit Interrupt Enable | 0: Overload frame transmit interrupt is disabled.<br>1: Overload frame transmit interrupt is enabled. | R/W |
| b14 | BLIE | Bus Lock Interrupt Enable | 0: Bus lock interrupt is disabled.<br>1: Bus lock interrupt is enabled. | R/W |
| b15 | ALIE | Arbitration Lost Interrupt Enable | 0: Arbitration lost interrupt is disabled.<br>1: Arbitration lost interrupt is enabled. | R/W |

**CHMDC[1:0] Bits (Mode Select)**

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see section 36.3.2, Channel Modes. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11b. When the CAN module has transitioned to channel halt mode depending on the setting of the CTRH.BOM[1:0] bits, the CHMDC[1:0] bits automatically become 10b.

**CSLPR Bit (Channel Stop Mode)**

Setting this bit to 1 places the channel in channel stop mode.

Setting this bit to 0 makes the channel leave from channel stop mode.

Do not modify this bit in channel communication mode or channel halt mode.

#### RTBO Bit (Forcible Return from Bus-off)

Setting this bit to 1 (forcible return from the bus off state) in the bus off state forcibly returns the state from the bus off state to the error active state. This bit is automatically set to 0. Setting this bit to 1 sets the STSH.TEC[7:0] and STSH.REC[7:0] flags to 00h and also sets the STSL.BOSTS flag to 0 (not in bus off state).The other registers remain unchanged. No bus off recovery interrupt request due to return from the bus off state is generated. Use this bit only when the CTRH.BOM[1:0] bits are 00b (ISO 11898-1 compliant).
A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the CAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

#### BEIE Bit (Protocol Error Interrupt Enable)

When the ERFLL.BEF flag becomes 1 while the BEIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

#### EWIE Bit (Error Warning Interrupt Enable)

When the ERFLL.EWF flag becomes 1 while the EWIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

#### EPIE Bit (Error Passive Interrupt Enable)

When the ERFLL.EPF flag becomes 1 while the EPIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

#### BOEIE Bit (Bus Off Entry Interrupt Enable)

When the ERFLL.BOEF flag becomes 1 while the BOEIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

#### BORIE Bit (Bus Off Recovery Interrupt Enable)

When the ERFLL.BORF flag becomes 1 while the BORIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

#### OLIE Bit (Overload Frame Transmit Interrupt Enable)

When the ERFLL.OVLF flag becomes 1 while the OLIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

#### BLIE Bit (Bus Lock Interrupt Enable)

When the ERFLL.BLF flag becomes 1 while the BLIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

#### ALIE Bit (Arbitration Lost Interrupt Enable)

When the ERFLL.ALF flag becomes 1 while the ALIE bit is 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

## 36.2.4 Control Register H (CTRH)

Address(es): RSCAN0.CTRH 000A 8306h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CTMS[1:0] | | CTME | ERRD | BOM[1:0] | | — | — | — | — | TAIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TAIE | Transmit Abort Interrupt Enable | 0: Transmit abort interrupt is disabled.<br>1: Transmit abort interrupt is enabled. | R/W |
| b4 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6, b5 | BOM[1:0] | Bus Off Recovery Mode Select | b6 b5<br>0 0: ISO 11898-1 compliant<br>0 1: Transition to channel halt mode at bus-off entry<br>1 0: Transition to channel halt mode at bus-off end<br>1 1: Transition to channel halt mode in the bus off state by a program request | R/W |
| b7 | ERRD | Error Display Mode Select | 0: Only the first error is indicated after b14 to b8 in the ERFLL register have all been cleared.<br>1: The error flags of all errors are indicated. | R/W |
| b8 | CTME | Communication Test Mode Enable | 0: Communication test mode is disabled.<br>1: Communication test mode is enabled. | R/W |
| b10, b9 | CTMS[1:0] | Communication Test Mode Select | b10 b9<br>0 0: Standard test mode<br>0 1: Listen-only mode<br>1 0: Self-test mode 0 (external loopback mode)<br>1 1: Self-test mode 1 (internal loopback mode) | R/W |
| b15 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### TAIE Bit (Transmit Abort Interrupt Enable)

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

### BOM[1:0] Bits (Bus Off Recovery Mode Select)

These bits are used to select a bus off recovery mode of the CAN module.

When the BOM[1:0] bits are set to 00b, return to the error active state from the bus off state is compliant with the ISO 11898-1 standard. That is, the CAN module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CTRL.CHMDC[1:0] bits are set to 10b (channel halt mode) before recessive bits are detected 128 times, the CAN module does not transition to channel halt mode until recessive bits are detected 128 times.

When the CAN module reaches the bus off state while the BOM[1:0] bits are set to 01b, the CTRL.CHMDC[1:0] bits are set to 10b and the CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the STSH.TEC[7:0] and STSH.REC[7:0] flags are set to 00h.

When the CAN module reaches the bus off state when the BOM[1:0] bits are set to 10b, the CTRL.CHMDC[1:0] bits are set to 10b and the CAN module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the STSH.TEC[7:0] and STSH.REC[7:0] flags are set to 00h.

When the BOM[1:0] bits are set to 11b and the CTRL.CHMDC[1:0] bits are set to 10b while the CAN module is in the bus off state, the CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the STSH.TEC[7:0] and STSH.REC[7:0] flags are set to 00h. However, if 11 consecutive recessive bits are detected 128 times and the CAN module has recovered to the error active state from the

bus off state before the CTRL.CHMDC[1:0] bits are set to 10b, a bus off recovery interrupt request is generated.
If the CPU requests transition to channel reset mode at the same time when the CAN module transitions to channel halt mode (at bus off entry when the BOM[1:0] bits are 01b or at bus off end when the BOM[1:0] bits are 10b), the CPU's request takes precedence. Modify these bits only in channel reset mode.

#### ERRD Bit (Error Display Mode Select)

This bit is used to control display mode of b14 to b8 in the ERFLL register.
When this bit is 0, only the flags of the first error become 1. If two or more errors occur first, all the flags of detected errors become 1.
When this bit is 1, all the flags of errors that have occurred become 1 regardless of the error occurrence order. Modify this bit only in channel reset mode or channel halt mode.

#### CTME Bit (Communication Test Mode Enable)

Setting this bit to 1 enables communication test mode. Modify this bit only in channel halt mode. This bit becomes 0 in channel reset mode.

#### CTMS[1:0] Bits (Communication Test Mode Select)

These bits are used to select a communication test mode. Modify these bits only in channel halt mode. These bits become 0 in channel reset mode.

## 36.2.5 Status Register L (STSL)

Address(es): RSCAN0.STSL 000A 8308h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | COMSTS | RECSTS | TRMSTS | BOSTS | EPSTS | CSLPSTS | CHLTSTS | CRSTSTS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CRSTSTS | Channel Reset Status Flag | 0: Not in channel reset mode<br>1: In channel reset mode | R |
| b1 | CHLTSTS | Channel Halt Status Flag | 0: Not in channel halt mode<br>1: In channel halt mode | R |
| b2 | CSLPSTS | Channel Stop Status Flag | 0: Not in channel stop mode<br>1: In channel stop mode | R |
| b3 | EPSTS | Error Passive Status Flag | 0: Not in error passive state<br>1: In error passive state | R |
| b4 | BOSTS | Bus Off Status Flag | 0: Not in bus off state<br>1: In bus off state | R |
| b5 | TRMSTS | Transmit Status Flag | 0: Bus idle or in reception<br>1: In transmission or bus off state | R |
| b6 | RECSTS | Receive Status Flag | 0: Bus idle, in transmission or bus off state<br>1: In reception | R |
| b7 | COMSTS | Communication Status Flag | 0: Communication is not ready.<br>1: Communication is ready. | R |
| b15 to b8 | — | Reserved | These bits are read as 0. | R |

**CRSTSTS Flag (Channel Reset Status Flag)**

This flag becomes 1 when the CAN module has transitioned to channel reset mode, and becomes 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 even if the CAN module transitions from channel reset mode to channel stop mode.

**CHLTSTS Flag (Channel Halt Status Flag)**

This flag becomes 1 when the CAN module has transitioned to channel halt mode, and becomes 0 when the CAN module has exited channel halt mode.

**CSLPSTS Flag (Channel Stop Status Flag)**

This flag becomes 1 when the CAN module has transitioned to channel stop mode, and becomes 0 when the CAN module has returned from channel stop mode.

**EPSTS Flag (Error Passive Status Flag)**

This flag becomes 1 when the CAN module has entered the error passive state (128 ≤ STSH.TEC[7:0] value ≤ 255 or 128 ≤ STSH.REC[7:0] value), and becomes 0 when the CAN module has exited the error passive state or has entered channel reset mode.

**BOSTS Flag (Bus Off Status Flag)**

This flag becomes 1 when the CAN module has entered the bus off state (STSH.TEC[7:0] value > 255), and becomes 0 when the CAN module has exited the bus off state.

**TRMSTS Flag (Transmit Status Flag)**

This flag becomes 1 when transmission has started, and becomes 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

**RECSTS Flag (Receive Status Flag)**

This flag becomes 1 when reception has started, and becomes 0 when the bus has become idle or transmission has started.

**COMSTS Flag (Communication Status Flag)**

This flag indicates that communication is ready.

This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag becomes 0 in channel reset mode or channel halt mode.

## 36.2.6 Status Register H (STSH)

Address(es): RSCAN0.STSH 000A 830Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TEC[7:0] | | | | | | | | REC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description | R/W |
|---|---|---|---|
| b7 to b0 | REC[7:0] | The receive error counter (REC) can be read. | R |
| b15 to b8 | TEC[7:0] | The transmit error counter (TEC) can be read. | R |

**REC[7:0] Flags**

These flags indicate the receive error counter value. For receive error counter increment/decrement conditions, see the CAN standard (ISO 11898-1).

These flags become 00h in channel reset mode.

**TEC[7:0] Flags**

These flags indicate the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN standard (ISO 11898-1).

These flags become 00h in channel reset mode.

## 36.2.7 Error Flag Register L (ERFLL)

Address(es): RSCAN0.ERFLL 000A 830Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | ADERR | B0ERR | B1ERR | CERR | AERR | FERR | SERR | ALF | BLF | OVLF | BORF | BOEF | EPF | EWF | BEF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | BEF | Bus Error Flag | 0: No channel bus error is detected.<br>1: Channel bus error is detected. | R/(W) *1 |
| b1 | EWF | Error Warning Flag | 0: No error warning is detected.<br>1: Error warning is detected. | R/(W) *1 |
| b2 | EPF | Error Passive Flag | 0: No error passive is detected.<br>1: Error passive is detected. | R/(W) *1 |
| b3 | BOEF | Bus Off Entry Flag | 0: No bus off entry is detected.<br>1: Bus off entry is detected. | R/(W) *1 |
| b4 | BORF | Bus Off Recovery Flag | 0: No bus off recovery is detected.<br>1: Bus off recovery is detected. | R/(W) *1 |
| b5 | OVLF | Overload Flag | 0: No overload is detected.<br>1: Overload is detected. | R/(W) *1 |
| b6 | BLF | Bus Lock Flag | 0: No channel bus lock is detected.<br>1: Channel bus lock is detected. | R/(W) *1 |
| b7 | ALF | Arbitration Lost Flag | 0: No arbitration lost is detected.<br>1: Arbitration lost is detected. | R/(W) *1 |
| b8 | SERR | Stuff Error Flag | 0: No stuff error is detected.<br>1: Stuff error is detected. | R/(W) *1 |
| b9 | FERR | Form Error Flag | 0: No form error is detected.<br>1: Form error is detected. | R/(W) *1 |
| b10 | AERR | ACK Error Flag | 0: No ACK error is detected.<br>1: ACK error is detected. | R/(W) *1 |
| b11 | CERR | CRC Error Flag | 0: No CRC error is detected.<br>1: CRC error is detected. | R/(W) *1 |
| b12 | B1ERR | Recessive Bit Error Flag | 0: No recessive bit error is detected.<br>1: Recessive bit error is detected. | R/(W) *1 |
| b13 | B0ERR | Dominant Bit Error Flag | 0: No dominant bit error is detected.<br>1: Dominant bit error is detected. | R/(W) *1 |
| b14 | ADERR | ACK Delimiter Error Flag | 0: No ACK delimiter error is detected.<br>1: ACK delimiter error is detected. | R/(W) *1 |
| b15 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

See the CAN standard (ISO 11898-1) if you want to check error occurrence conditions. To clear each flag of this register, write 0 by the program. These flags cannot be set to 1 by the program. If any of these flags becomes 1 at the timing when the program writes 0 to the flag, the flag becomes 1. Each flag becomes 0 in channel reset mode.
With respect to b14 to b8 in the ERFLL register, if an error is detected with all flags of b14 to b8 set to 0 when the CTRH.ERRD bit is set to 0 (only the first error information is displayed), the corresponding flag becomes 1.

**BEF Flag (Bus Error Flag)**

This flag becomes 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags becomes 1.

#### EWF Flag (Error Warning Flag)

This flag becomes 1 only when the STSH.REC[7:0] or STSH.TEC[7:0] value exceeds 95 for the first time. Therefore, if the program writes 0 to this flag with the STSH.REC[7:0] or STSH.TEC [7:0] value remaining over 95, this bit is not set to 1 until both STSH.REC[7:0] and STSH.TEC[7:0] values become 95 or less and then the STSH.REC[7:0] or STSH.TEC[7:0] value exceeds 95 again.

#### EPF Flag (Error Passive Flag)

This flag becomes 1 when the CAN module becomes error passive state (STSH.REC[7:0] or STSH.TEC[7:0] value > 127). This flag becomes 1 only when the STSH.REC[7:0] or STSH.TEC[7:0] value exceeds 127 for the first time. Therefore, if the program writes 0 to this flag with the STSH.REC[7:0] or STSH.TEC[7:0] value remaining over 127, this bit is not set to 1 until both STSH.REC[7:0] and STSH.TEC[7:0] values become 127 or less and then the STSH.REC[7:0] or STSH.TEC[7:0] value exceeds 127 again.

#### BOEF Flag (Bus Off Entry Flag)

This flag becomes 1 when the state becomes bus off state (STSH.TEC[7:0] value > 255). This flag also becomes 1 when the state becomes bus off state with the CTRH.BOM[1:0] bits set to 01b (transition to channel halt mode at bus off entry).

#### BORF Flag (Bus Off Recovery Flag)

This flag becomes 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CTRL.CHMDC[1:0] bits are set to 01b (channel reset mode).
- The CTRL.RTBO bit is set to 1 (forcible return from the bus off state is made).
- The CTRH.BOM[1:0] bits are set to 01b (transition to channel halt mode at bus off entry).
- The CTRL.CHMDC[1:0] bits are set to 10b (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the CTRH.BOM[1:0] bits set to 11b (transition to channel halt mode upon a request from the program during bus off).

#### OVLF Flag (Overload Flag)

This flag becomes 1 when the overload frame transmit condition has been detected when performing reception or transmission.

#### BLF Flag (Bus Lock Flag)

This flag becomes 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of the bus lock becomes possible again if either of the following conditions is met.

- A recessive bit is detected after the BLF flag has been modified from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF flag has been modified from 1 to 0.

#### ALF Flag (Arbitration Lost Flag)

This flag becomes 1 when an arbitration lost has been detected.

#### SERR Flag (Stuff Error Flag)

This flag becomes 1 when a stuff error has been detected.

#### FERR Flag (Form Error Flag)

This flag becomes 1 when a form error has been detected.

**AERR Flag (ACK Error Flag)**

This flag becomes 1 when an ACK error has been detected.

**CERR Flag (CRC Error Flag)**

This flag becomes 1 when a CRC error has been detected.

**B1ERR Flag (Recessive Bit Error Flag)**

This flag becomes 1 when a dominant bit has been detected though a recessive bit was transmitted.

**B0ERR Flag (Dominant Bit Error Flag)**

This flag becomes 1 when a recessive bit has been detected though a dominant bit was transmitted.

**ADERR Flag (ACK Delimiter Error Flag)**

This flag becomes 1 when a form error has been detected in the ACK delimiter during transmission.

## 36.2.8 Error Flag Register H (ERFLH)

Address(es): RSCAN0.ERFLH 000A 830Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | CRCREG[14:0] | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b14 to b0 | CRCREG[14:0] | CRC Calculation Data | A CRC value calculated based on the transmit message or receive message is indicated. | R |
| b15 | — | Reserved | This bit is read as 0. | R |

**CRCREG[14:0] Bits (CRC Calculation Data)**

When the CTRH.CTME bit is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTRH.CTME bit is set to 0 (communication test mode is disabled), these bits are read as 0.

## 36.2.9 Global Configuration Register L (GCFGL)

Address(es): RSCAN.GCFGL 000A 8322h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | TSSS | TSP[3:0] | | | | — | — | — | DCS | MME | DRE | DCE | TPRI |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TPRI | Transmit Priority Select | 0: ID priority<br>1: Transmit buffer number priority | R/W |
| b1 | DCE | DLC Check Enable | 0: DLC check is disabled.<br>1: DLC check is enabled. | R/W |
| b2 | DRE | DLC Replacement Enable | 0: DLC replacement is disabled.<br>1: DLC replacement is enabled. | R/W |
| b3 | MME | Mirror Function Enable | 0: Mirror function is disabled.<br>1: Mirror function is enabled. | R/W |
| b4 | DCS | CAN Clock Source Select | 0: PCLK<br>1: CANMCLK (obtained from the main clock) | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 to b8 | TSP[3:0] | Timestamp Clock Source Division | b11 b10 b9 b8<br>0 0 0 0: Not divided<br>0 0 0 1: Divided by 2<br>0 0 1 0: Divided by 4<br>0 0 1 1: Divided by 8<br>0 1 0 0: Divided by 16<br>0 1 0 1: Divided by 32<br>0 1 1 0: Divided by 64<br>0 1 1 1: Divided by 128<br>1 0 0 0: Divided by 256<br>1 0 0 1: Divided by 512<br>1 0 1 0: Divided by 1024<br>1 0 1 1: Divided by 2048<br>1 1 0 0: Divided by 4096<br>1 1 0 1: Divided by 8192<br>1 1 1 0: Divided by 16384<br>1 1 1 1: Divided by 32768 | R/W |
| b12 | TSSS | Timestamp Clock Source Select | 0: PCLK<br>1: CAN bit time clock | R/W |
| b15 to b13 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Modify the GCFGL register only in global reset mode.

### TPRI Bit (Transmit Priority Select)

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO 11898-1 standard). When this bit is set to 1, transmit buffer number priority is selected and the minimum number of transmit buffer specified for transmission takes precedence.

### DCE Bit (DLC Check Enable)

Setting this bit to 1 makes the DLC check function available. Set the GAFLPHj.GAFLDLC[3:0] bits to 0000b before setting the DCE bit to 0.

**DRE Bit (DLC Replacement Enable)**

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00h is stored in the data byte that exceeds the DLC value of the receive rule.
When the DCE bit is set to 1 (DLC check is enabled), the DLC replacement function is available.

**MME Bit (Mirror Function Enable)**

Setting this bit to 1 makes the mirror function available.

**DCS Bit (CAN Clock Source Select)**

When this bit is set to 0, the peripheral clock (PCLK) divided by 2 is used as the CAN clock source (fCAN).
When this bit is set to 1, CANMCLK obtained from the EXTAL pin is used as the CAN clock source (fCAN).

**TSP[3:0] Bits (Timestamp Clock Source Division)**

The clock obtained by dividing the clock source selected by the TSSS bit by the TSP[3:0] value is the count source of the timestamp counter.

**TSSS Bit (Timestamp Clock Source Select)**

This bit is used to select a clock source of the timestamp counter.

## 36.2.10 Global Configuration Register H (GCFGH)

Address(es): RSCAN.GCFGH 000A 8324h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ITRCP[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | ITRCP[15:0] | Interval Timer Prescaler Set | If the set value is M, PCLK is frequency-divided by M.<br>Setting 0000h is prohibited when the interval timer is in use. | R/W |

Modify the GCFGH register only in global reset mode.

**ITRCP[15:0] Bits (Interval Timer Prescaler Set)**

These bits are used to set a clock source division value of the interval timer for FIFO buffers. For details, see section 36.5.3 (1) Interval Transmission Function.

## 36.2.11 Global Control Register L (GCTRL)

Address(es): RSCAN.GCTRL 000A 8326h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | THLEIE | MEIE | DEIE | — | — | — | — | — | GSLPR | GMDC[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | GMDC[1:0] | Global Mode Select | b1 b0<br>0 0: Global operating mode<br>0 1: Global reset mode<br>1 0: Global test mode<br>1 1: Setting prohibited | R/W |
| b2 | GSLPR | Global Stop Mode | 0: Other than global stop mode<br>1: Global stop mode | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | DEIE | DLC Error Interrupt Enable | 0: DLC error interrupt is disabled.<br>1: DLC error interrupt is enabled. | R/W |
| b9 | MEIE | FIFO Message Lost Interrupt Enable | 0: FIFO message lost interrupt is disabled.<br>1: FIFO message lost interrupt is enabled. | R/W |
| b10 | THLEIE | Transmit History Buffer Overflow Interrupt Enable | 0: Transmit history buffer overflow interrupt is disabled.<br>1: Transmit history buffer overflow interrupt is enabled. | R/W |
| b15 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### GMDC[1:0] Bits (Global Mode Select)

These bits are used to select the mode of entire CAN module (global operating mode, global reset mode, or global test mode). For details, see section 36.3.1, Global Modes. Setting the GSLPR bit to 1 in global reset mode places the CAN module in global stop mode.

### GSLPR Bit (Global Stop Mode)

Setting this bit to 1 places the CAN module in global stop mode.
Setting this bit to 0 makes the CAN module leave from global stop mode.
Do not modify this bit in global operating mode or in global test mode.

### DEIE Bit (DLC Error Interrupt Enable)

When the GERFLL.DEF flag becomes 1 while the DEIE bit is 1, an interrupt request is generated. Modify this bit only in global reset mode.

### MEIE Bit (FIFO Message Lost Interrupt Enable)

When the GERFLL.MES flag becomes 1 while the MEIE bit is 1, an interrupt request is generated. Modify this bit only in global reset mode.

### THLEIE Bit (Transmit History Buffer Overflow Interrupt Enable)

When the GERFLL.THLES flag becomes 1 while the THLEIE bit is 1, an interrupt request is generated. Modify this bit only in global reset mode.

## 36.2.12 Global Control Register H (GCTRH)

Address(es): RSCAN.GCTRH 000A 8328h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSRST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TSRST | Timestamp Counter Reset | Setting the TSRST bit to 1 resets the timestamp counter. This bit is read as 0. | R/W |
| b15 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### TSRST Bit (Timestamp Counter Reset)

This bit is used to reset the timestamp counter. When this bit is set to 1, the GTSC register is set to 0000h.

## 36.2.13 Global Status Register (GSTS)

Address(es): RSCAN.GSTS 000A 832Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | GRAMINIT | GSLPSTS | GHLTSTS | GRSTSTS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | GRSTSTS | Global Reset Status Flag | 0: Not in global reset mode<br>1: In global reset mode | R |
| b1 | GHLTSTS | Global Test Status Flag | 0: Not in global test mode<br>1: In global test mode | R |
| b2 | GSLPSTS | Global Stop Status Flag | 0: Not in global stop mode<br>1: In global stop mode | R |
| b3 | GRAMINIT | CAN RAM Initialization Status Flag | 0: CAN RAM initialization is completed.<br>1: CAN RAM initialization is ongoing. | R |
| b15 to b4 | — | Reserved | These bits are read as 0. | R |

### GRSTSTS Flag (Global Reset Status Flag)

This flag becomes 1 when the CAN module has transitioned to global reset mode, and becomes 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

### GHLTSTS Flag (Global Test Status Flag)

This flag becomes 1 when the CAN module has transitioned to global test mode, and becomes 0 when the CAN module has exited global test mode.

### GSLPSTS Flag (Global Stop Status Flag)

This flag becomes 1 when the CAN module has transitioned to global stop mode, and becomes 0 when the CAN module

has returned from global stop mode.

**GRAMINIT Flag (CAN RAM Initialization Status Flag)**

This flag indicates the initialization status of the CAN RAM.
This flag becomes 1 after the CAN module is enabled, and becomes 0 when CAN RAM initialization is completed.

## 36.2.14 Global Error Flag Register (GERFLL)

Address(es): RSCAN.GERFLL 000A 832Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | THLES | MES | DEF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | DEF | DLC Error Flag | 0: No DLC error is present.<br>1: A DLC error is present. | R/(W) *1 |
| b1 | MES | FIFO Message Lost Status Flag | 0: No FIFO message lost error is present.<br>1: A FIFO message lost error is present. | R |
| b2 | THLES | Transmit History Buffer Overflow Status Flag | 0: No transmit history buffer overflow is present.<br>1: A transmit history buffer overflow is present. | R |
| b7 to b3 | — | Reserved | The read value is undefined. The write value should be 0. | R/W |

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

All flags in the GERFLL register become 0 in global reset mode.

**DEF Flag (DLC Error Flag)**

The DEF flag becomes 1 when an error has been detected during the DLC check. This flag can be set to 0 by writing 0 by the program.

**MES Flag (FIFO Message Lost Status Flag)**

The MES flag becomes 1 when any one of the RFSTSm.RFMLT flags or the CFSTS0.CFMLT flag becomes 1.
This flag becomes 0 when all RFSTSm.RFMLT flags and the CFSTS0.CFMLT flag are set to 0.

**THLES Flag (Transmit History Buffer Overflow Status Flag)**

The THLES flag becomes 1 when the THLSTS0.THLELT flag becomes 1.
This flag becomes 0 when the THLSTS0.THLELT flag is set to 0.

## 36.2.15 Global Transmit Interrupt Status Register (GTINTSTS)

Address(es): RSCAN.GTINTSTS 000A 8388h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | THIF0 | CFTIF0 | TAIF0 | TSIF0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TSIF0 | RSCAN0 Transmit Buffer Interrupt Status Flag | 0: No transmit buffer transmit complete interrupt request is present.<br>1: A transmit buffer transmit complete interrupt request is present. | R |
| b1 | TAIF0 | RSCAN0 Transmit Buffer Abort Interrupt Status Flag | 0: No transmit buffer abort interrupt request is present.<br>1: A transmit buffer abort interrupt request is present. | R |
| b2 | CFTIF0 | RSCAN0 Transmit/Receive FIFO Interrupt Status Flag | 0: No transmit/receive FIFO transmit interrupt request is present.<br>1: A transmit/receive FIFO transmit interrupt request is present. | R |
| b3 | THIF0 | RSCAN0 Transmit History Interrupt Status Flag | 0: No transmit history interrupt request is present.<br>1: A transmit history interrupt request is present. | R |
| b15 to b4 | — | Reserved | These bits are read as 0. | R |

All flags in the GTINTSTS register become 0 in global reset or channel reset mode.

**TSIF0 Flag (RSCAN0 Transmit Buffer Interrupt Status Flag)**

The TSIF0 flag becomes 1 when the TMIEC.TMIEp bit is set to 1 (enabling interrupts) and the corresponding TMSTSp.TMTRF[1:0] flags become 10b (transmission has been completed without transmit abort request) or 11b (transmission has been completed with transmit abort request).
This flag becomes 0 when all TMSTSp.TMTRF[1:0] flags that satisfy a condition for setting the TSIF0 flag to 1 are set to 00b. This flag also becomes 0 when the TMIEC.TMIEp bit is set to 0.

**TAIF0 Flag (RSCAN0 Transmit Buffer Abort Interrupt Status Flag)**

The TAIF0 flag becomes 1 when the CTRH.TAIE bit is set to 1 (enabling interrupts) and the TMSTSp.TMTRF[1:0] flags become 01b (transmit abort has been completed).
This flag becomes 0 when the TMSTSp.TMTRF[1:0] flags, which indicate that the abort of transmission has been completed, are set to 00b.

**CFTIF0 Flag (RSCAN0 Transmit/Receive FIFO Interrupt Status Flag)**

The CFTIF0 flag becomes 1 when the CFCCL0.CFTXIE bit is set to 1 (enabling interrupts) and the CFSTS0.CFTXIF flag becomes 1 (interrupt request present).
This flag becomes 0 when the CFSTS0.CFTXIF flag is set to 0. This flag also becomes 0 when the CFCCL0.CFTXIE bit is set to 0.

**THIF0 Flag (RSCAN0 Transmit History Interrupt Status Flag)**

The THIF0 flag becomes 1 when the THLCC0.THLIE bit is set to 1 (enabling interrupts) and the THLSTS0.THLIF flag becomes 1 (interrupt request present).
This flag becomes 0 when the THLSTS0.THLIF flag is set to 0. This flag also becomes 0 when the THLCC0.THLIE bit is set to 0.

## 36.2.16 Timestamp Register (GTSC)

Address(es): RSCAN.GTSC 000A 832Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description | Counter Value | R/W |
|---|---|---|---|---|
| b15 to b0 | TS[15:0] | The timestamp counter value can be read. | 0000h to FFFFh | R |

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. The TS[15:0] value is captured when the SOF is detected and then stored in the receive buffer or the FIFO buffer.
The timestamp counter is initialized in global reset mode.
The timestamp counter start timing and stop timing depend on the count source.

- When the GCFGL.TSSS value is 0 (PCLK is selected):
  The timestamp counter starts counting when the CAN module has transitioned to global operating mode.
  This counter stops counting when the CAN module has transitioned to global stop mode or global test mode.
- When the GCFGL.TSSS value is 1 (CAN bit time clock is selected):
  The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
  This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

## 36.2.17 Receive Rule Number Configuration Register (GAFLCFG)

Address(es): RSCAN.GAFLCFG 000A 8330h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | RNC0[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | RNC0[4:0] | RSCAN0 Receive Rule Number Set | Set the number of receive rules of channel 0.<br>Set these bits to a value within a range of 00h to 10h. | R/W |
| b15 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Modify the GAFLCFG register only in global reset mode.
Up to 16 rules can be registered in the receive rule table.

### RNC0[4:0] Bits (RSCAN0 Receive Rule Number Set)

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.
Set these bits to a value within a range of 00h to 10h.

## 36.2.18 Receive Rule Entry Register jAL (GAFLIDLj) (j = 0 to 15)

Address(es): RSCAN.GAFLIDL0 000A 83A0h, RSCAN.GAFLIDL1 000A 83ACh, RSCAN.GAFLIDL2 000A 83B8h, RSCAN.GAFLIDL3 000A 83C4h, RSCAN.GAFLIDL4 000A 83D0h, RSCAN.GAFLIDL5 000A 83DCh, RSCAN.GAFLIDL6 000A 83E8h, RSCAN.GAFLIDL7 000A 83F4h, RSCAN.GAFLIDL8 000A 8400h, RSCAN.GAFLIDL9 000A 840Ch, RSCAN.GAFLIDL10 000A 8418h, RSCAN.GAFLIDL11 000A 8424h, RSCAN.GAFLIDL12 000A 8430h, RSCAN.GAFLIDL13 000A 843Ch, RSCAN.GAFLIDL14 000A 8448h, RSCAN.GAFLIDL15 000A 8454h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLID[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | GAFLID[15:0] | ID Set L | Set the ID of the receive rule.<br>For the standard ID, set the ID in b10 to b0 and set b15 to b11 to 0. | R/W |

Modify the GAFLIDLj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

**GAFLID[15:0] Bits (ID Set L)**

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID in the received message during the acceptance filter processing.

## 36.2.19 Receive Rule Entry Register jAH (GAFLIDHj) (j = 0 to 15)

Address(es): RSCAN.GAFLIDH0 000A 83A2h, RSCAN.GAFLIDH1 000A 83AEh, RSCAN.GAFLIDH2 000A 83BAh, RSCAN.GAFLIDH3 000A 83C6h, RSCAN.GAFLIDH4 000A 83D2h, RSCAN.GAFLIDH5 000A 83DEh, RSCAN.GAFLIDH6 000A 83EAh, RSCAN.GAFLIDH7 000A 83F6h, RSCAN.GAFLIDH8 000A 8402h, RSCAN.GAFLIDH9 000A 840Eh, RSCAN.GAFLIDH10 000A 841Ah, RSCAN.GAFLIDH11 000A 8426h, RSCAN.GAFLIDH12 000A 8432h, RSCAN.GAFLIDH13 000A 843Eh, RSCAN.GAFLIDH14 000A 844Ah, RSCAN.GAFLIDH15 000A 8456h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLIDE | GAFLRTR | GAFLLB | GAFLID[28:16] | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b12 to b0 | GAFLID[28:16] | ID Set H | Set the ID of the receive rule.<br>For the standard ID, set these bits to 0. | R/W |
| b13 | GAFLLB | Receive Rule Target Message Select | 0: When a message transmitted from another CAN node is received<br>1: When a message transmitted from own node is received | R/W |
| b14 | GAFLRTR | RTR Select | 0: Data frame<br>1: Remote frame | R/W |
| b15 | GAFLIDE | IDE Select | 0: Standard ID<br>1: Extended ID | R/W |

Modify the GAFLIDHj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

### GAFLID[28:16] Bits (ID Set H)

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID in the received message during the acceptance filter processing.

### GAFLLB Bit (Receive Rule Target Message Select)

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when receiving messages transmitted from the own CAN node.

### GAFLRTR Bit (RTR Select)

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

### GAFLIDE Bit (IDE Select)

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

## 36.2.20 Receive Rule Entry Register jBL (GAFLMLj) (j = 0 to 15)

Address(es): RSCAN.GAFLML0 000A 83A4h, RSCAN.GAFLML1 000A 83B0h, RSCAN.GAFLML2 000A 83BCh, RSCAN.GAFLML3 000A 83C8h, RSCAN.GAFLML4 000A 83D4h, RSCAN.GAFLML5 000A 83E0h, RSCAN.GAFLML6 000A 83ECh, RSCAN.GAFLML7 000A 83F8h, RSCAN.GAFLML8 000A 8404h, RSCAN.GAFLML9 000A 8410h, RSCAN.GAFLML10 000A 841Ch, RSCAN.GAFLML11 000A 8428h, RSCAN.GAFLML12 000A 8434h, RSCAN.GAFLML13 000A 8440h, RSCAN.GAFLML14 000A 844Ch, RSCAN.GAFLML15 000A 8458h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLIDM[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | GAFLIDM[15:0] | ID Mask L | 0: The corresponding ID bit is not compared.<br>1: The corresponding ID bit is compared. | R/W |

Modify the GAFLMLj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

**GAFLIDM[15:0] Bits (ID Mask L)**

These bits are used to mask the corresponding ID bit of the receive rule.

## 36.2.21 Receive Rule Entry Register jBH (GAFLMHj) (j = 0 to 15)

Address(es): RSCAN.GAFLMH0 000A 83A6h, RSCAN.GAFLMH1 000A 83B2h, RSCAN.GAFLMH2 000A 83BEh, RSCAN.GAFLMH3 000A 83CAh, RSCAN.GAFLMH4 000A 83D6h, RSCAN.GAFLMH5 000A 83E2h, RSCAN.GAFLMH6 000A 83EEh, RSCAN.GAFLMH7 000A 83FAh, RSCAN.GAFLMH8 000A 8406h, RSCAN.GAFLMH9 000A 8412h, RSCAN.GAFLMH10 000A 841Eh, RSCAN.GAFLMH11 000A 842Ah, RSCAN.GAFLMH12 000A 8436h, RSCAN.GAFLMH13 000A 8442h, RSCAN.GAFLMH14 000A 844Eh, RSCAN.GAFLMH15 000A 845Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLIDEM | GAFLRTRM | — | GAFLIDM[28:16] | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b12 to b0 | GAFLIDM[28:16] | ID Mask H | 0: The corresponding ID bit is not compared.<br>1: The corresponding ID bit is compared. | R/W |
| b13 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b14 | GAFLRTRM | RTR Mask | 0: The RTR bit is not compared.<br>1: The RTR bit is compared | R/W |
| b15 | GAFLIDEM | IDE Mask | 0: The IDE bit is not compared.<br>1: The IDE bit is compared. | R/W |

Modify the GAFLMHj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

### GAFLIDM[28:16] Bits (ID Mask H)

These bits are used to mask the corresponding ID bit of the receive rule.

### GAFLRTRM Bit (RTR Mask)

This bit is used to mask the RTR bit of the receive rule.

### GAFLIDEM Bit (IDE Mask)

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDHj.GAFLIDE bit.

When this bit is set to 0, it is regarded that all received messages have matched the specified ID format. To set the GAFLIDEM bit to 0, set the GAFLMHj.GAFLIDM[28:16] bits and the GAFLMLj.GAFLIDM[15:0] bits to all 0s.

## 36.2.22 Receive Rule Entry Register jCL (GAFLPLj) (j = 0 to 15)

Address(es): RSCAN.GAFLPL0 000A 83A8h, RSCAN.GAFLPL1 000A 83B4h, RSCAN.GAFLPL2 000A 83C0h, RSCAN.GAFLPL3 000A 83CCh, RSCAN.GAFLPL4 000A 83D8h, RSCAN.GAFLPL5 000A 83E4h, RSCAN.GAFLPL6 000A 83F0h, RSCAN.GAFLPL7 000A 83FCh, RSCAN.GAFLPL8 000A 8408h, RSCAN.GAFLPL9 000A 8414h, RSCAN.GAFLPL10 000A 8420h, RSCAN.GAFLPL11 000A 842Ch, RSCAN.GAFLPL12 000A 8438h, RSCAN.GAFLPL13 000A 8444h, RSCAN.GAFLPL14 000A 8450h, RSCAN.GAFLPL15 000A 845Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLRMV | GAFLRMDP[6:0] | | | | | | | — | — | — | GAFLFDP4 | — | — | GAFLFDP1 | GAFLFDP0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | GAFLFDP0 | Receive FIFO Buffer Select 0 | 0: Not select a receive FIFO buffer 0<br>1: Select a receive FIFO buffer 0 | R/W |
| b1 | GAFLFDP1 | Receive FIFO Buffer Select 1 | 0: Not select a receive FIFO buffer 1<br>1: Select a receive FIFO buffer 1 | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | GAFLFDP4 | RSCAN0 Transmit/Receive FIFO Buffer Select 0 | 0: Not select an RSCAN0 transmit/receive FIFO buffer 0<br>1: Select an RSCAN0 transmit/receive FIFO buffer 0 | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b14 to b8 | GAFLRMDP[6:0] | Receive Buffer Number Select | Set the receive buffer number to store receive messages. | R/W |
| b15 | GAFLRMV | Receive Buffer Enable | 0: No receive buffer is used.<br>1: A receive buffer is used. | R/W |

Modify the GAFLPLj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

**GAFLFDP0 Bit (Receive FIFO Buffer Select 0),**
**GAFLFDP1 Bit (Receive FIFO Buffer Select 1),**
**GAFLFDP4 Bit (RSCAN0 Transmit/Receive FIFO Buffer Select 0)**

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to two FIFO buffers are selectable. However, when the GAFLPLj.GAFLRMV bit is set to 1 (a receive buffer is used), up to one FIFO buffer is selectable. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFCCH0.CFM[1:0] bits are set to 00b (receive mode) are selectable.

**GAFLRMDP[6:0] Bits (Receive Buffer Number Select)**

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the RMNB.NRXMB[4:0] bits.

**GAFLRMV Bit (Receive Buffer Enable)**

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

## 36.2.23 Receive Rule Entry Register jCH (GAFLPHj) (j = 0 to 15)

Address(es): RSCAN.GAFLPH0 000A 83AAh, RSCAN.GAFLPH1 000A 83B6h, RSCAN.GAFLPH2 000A 83C2h, RSCAN.GAFLPH3 000A 83CEh, RSCAN.GAFLPH4 000A 83DAh, RSCAN.GAFLPH5 000A 83E6h, RSCAN.GAFLPH6 000A 83F2h, RSCAN.GAFLPH7 000A 83FEh, RSCAN.GAFLPH8 000A 840Ah, RSCAN.GAFLPH9 000A 8416h, RSCAN.GAFLPH10 000A 8422h, RSCAN.GAFLPH11 000A 842Eh, RSCAN.GAFLPH12 000A 843Ah, RSCAN.GAFLPH13 000A 8446h, RSCAN.GAFLPH14 000A 8452h, RSCAN.GAFLPH15 000A 845Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLDLC[3:0] | | | | GAFLPTR[11:0] | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b11 to b0 | GAFLPTR[11:0] | Receive Rule Label | Set the 12-bit label information. | R/W |
| b15 to b12 | GAFLDLC[3:0] | Receive Rule DLC | b15 b12<br>0 0 0 0: 0 or more data bytes (DLC check is disabled)<br>0 0 0 1: 1 or more data bytes<br>0 0 1 0: 2 or more data bytes<br>0 0 1 1: 3 or more data bytes<br>0 1 0 0: 4 or more data bytes<br>0 1 0 1: 5 or more data bytes<br>0 1 1 0: 6 or more data bytes<br>0 1 1 1: 7 or more data bytes<br>1 x x x: 8 or more data bytes | R/W |

x: Don’t care

Modify the GAFLPHj register only when the GRWCR.RPAGE bit is set to 0 in global reset mode.

### GAFLPTR[11:0] Bits (Receive Rule Label)

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

### GAFLDLC[3:0] Bits (Receive Rule DLC)

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000b disables the DLC check function allowing messages with any data length to pass the DLC check.

## 36.2.24 Receive Buffer Number Configuration Register (RMNB)

Address(es): RSCAN.RMNB 000A 8332h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | NRXMB[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | NRXMB[4:0] | Receive Buffer Number Configuration | Set the number of receive buffers.<br>Set a value of 0 to 16. | R/W |
| b15 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Modify the RMNB register only in global reset mode.

### NRXMB[4:0] Bits (Receive Buffer Number Configuration)

These bits are used to set the total number of receive buffers of the CAN module. The maximum value is 16.
Setting these bits to all 0s makes receive buffers unavailable.

## 36.2.25 Receive Buffer Receive Complete Flag Register (RMND0)

Address(es): RSCAN.RMND0 000A 8334h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMNS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | RMNS[15:0] | Receive Buffer Receive Complete Flag n | 0: Receive buffer n contains no new message (n = 0 to 15).<br>1: Receive buffer n contains a new message. | R/W |

Write 0 to the RMND0 register in global operating mode or global test mode.

### RMNS[15:0] Flags (Receive Buffer Receive Complete Flag n)

Each of the RMNS[15:0] flags becomes 1 when the processing for storing a message in the corresponding receive buffer starts.
To set these flags to 0, write 0 to the corresponding flag by the program. In this case, write this register in 16-bit unit to ensure that only the specified bit is set to 0 and the other bits are set to 1. These bits cannot be set to 0 while a message is being stored. It takes time of 10 clock cycles of PCLK for storing a message.
These flags become 0 in global reset mode.

## 36.2.26 Receive Buffer Register nAL (RMIDLn) (n = 0 to 15)

Address(es): RSCAN.RMIDL0 000A 83A0h, RSCAN.RMIDL1 000A 83B0h, RSCAN.RMIDL2 000A 83C0h, RSCAN.RMIDL3 000A 83D0h, RSCAN.RMIDL4 000A 83E0h, RSCAN.RMIDL5 000A 83F0h, RSCAN.RMIDL6 000A 8400h, RSCAN.RMIDL7 000A 8410h, RSCAN.RMIDL8 000A 8420h, RSCAN.RMIDL9 000A 8430h, RSCAN.RMIDL10 000A 8440h, RSCAN.RMIDL11 000A 8450h, RSCAN.RMIDL12 000A 8460h, RSCAN.RMIDL13 000A 8470h, RSCAN.RMIDL14 000A 8480h, RSCAN.RMIDL15 000A 8490h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMID[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | RMID[15:0] | Receive Buffer ID Data L | The standard ID or extended ID of received message can be read.<br>Read bits 10 to 0 for standard ID. Bits 15 to 11 are read as 0. | R |

This register can be read when the GRWCR.RPAGE bit is 1.

**RMID[15:0] Bits (Receive Buffer ID Data L)**

These bits indicate the ID of the message stored in the receive buffer.

## 36.2.27 Receive Buffer Register nAH (RMIDHn) (n = 0 to 15)

Address(es): RSCAN.RMIDH0 000A 83A2h, RSCAN.RMIDH1 000A 83B2h, RSCAN.RMIDH2 000A 83C2h, RSCAN.RMIDH3 000A 83D2h, RSCAN.RMIDH4 000A 83E2h, RSCAN.RMIDH5 000A 83F2h, RSCAN.RMIDH6 000A 8402h, RSCAN.RMIDH7 000A 8412h, RSCAN.RMIDH8 000A 8422h, RSCAN.RMIDH9 000A 8432h, RSCAN.RMIDH10 000A 8442h, RSCAN.RMIDH11 000A 8452h, RSCAN.RMIDH12 000A 8462h, RSCAN.RMIDH13 000A 8472h, RSCAN.RMIDH14 000A 8482h, RSCAN.RMIDH15 000A 8492h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMIDE | RMRTR | — | RMID[28:16] | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b12 to b0 | RMID[28:16] | Receive Buffer ID Data H | The standard ID or extended ID of received message can be read.<br>For standard ID, these bits are read as 0. | R |
| b13 | — | Reserved | This bit is read as 0. | R |
| b14 | RMRTR | Receive Buffer RTR | 0: Data frame<br>1: Remote frame | R |
| b15 | RMIDE | Receive Buffer IDE | 0: Standard ID<br>1: Extended ID | R |

This register can be read when the GRWCR.RPAGE bit is 1.

### RMID[28:16] Bits (Receive Buffer ID Data H)

These bits indicate the ID of the message stored in the receive buffer.

### RMRTR Bit (Receive Buffer RTR)

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

### RMIDE Bit (Receive Buffer IDE)

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

## 36.2.28 Receive Buffer Register nBL (RMTSn) (n = 0 to 15)

Address(es): RSCAN.RMTS0 000A 83A4h, RSCAN.RMTS1 000A 83B4h, RSCAN.RMTS2 000A 83C4h, RSCAN.RMTS3 000A 83D4h, RSCAN.RMTS4 000A 83E4h, RSCAN.RMTS5 000A 83F4h, RSCAN.RMTS6 000A 8404h, RSCAN.RMTS7 000A 8414h, RSCAN.RMTS8 000A 8424h, RSCAN.RMTS9 000A 8434h, RSCAN.RMTS10 000A 8444h, RSCAN.RMTS11 000A 8454h, RSCAN.RMTS12 000A 8464h, RSCAN.RMTS13 000A 8474h, RSCAN.RMTS14 000A 8484h, RSCAN.RMTS15 000A 8494h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | RMTS[15:0] | Receive Buffer Timestamp Data | Timestamp value of the received message can be read. | R |

This register can be read when the GRWCR.RPAGE bit is 1.

**RMTS[15:0] Bits (Receive Buffer Timestamp Data)**

These bits indicate the timestamp value of the message stored in the receive buffer.

## 36.2.29 Receive Buffer Register nBH (RMPTRn) (n = 0 to 15)

Address(es): RSCAN.RMPTR0 000A 83A6h, RSCAN.RMPTR1 000A 83B6h, RSCAN.RMPTR2 000A 83C6h, RSCAN.RMPTR3 000A 83D6h, RSCAN.RMPTR4 000A 83E6h, RSCAN.RMPTR5 000A 83F6h, RSCAN.RMPTR6 000A 8406h, RSCAN.RMPTR7 000A 8416h, RSCAN.RMPTR8 000A 8426h, RSCAN.RMPTR9 000A 8436h, RSCAN.RMPTR10 000A 8446h, RSCAN.RMPTR11 000A 8456h, RSCAN.RMPTR12 000A 8466h, RSCAN.RMPTR13 000A 8476h, RSCAN.RMPTR14 000A 8486h, RSCAN.RMPTR15 000A 8496h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDLC[3:0] | | | | RMPTR[11:0] | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b11 to b0 | RMPTR[11:0] | Receive Buffer Label Data | Label information of the received message can be read. | R |
| b15 to b12 | RMDLC[3:0] | Receive Buffer DLC Data | b15 b12<br>0 0 0 0: 0 data bytes<br>0 0 0 1: 1 data byte<br>0 0 1 0: 2 data bytes<br>0 0 1 1: 3 data bytes<br>0 1 0 0: 4 data bytes<br>0 1 0 1: 5 data bytes<br>0 1 1 0: 6 data bytes<br>0 1 1 1: 7 data bytes<br>1 x x x: 8 data bytes | R |

x: Don’t care

This register can be read when the GRWCR.RPAGE bit is 1.

### RMPTR[11:0] Bits (Receive Buffer Label Data)

These bits indicate the label information of the message stored in the receive buffer.

### RMDLC[3:0] Bits (Receive Buffer DLC Data)

These bits indicate the data length of the message stored in the receive buffer.

## 36.2.30 Receive Buffer Register nCL (RMDF0n) (n = 0 to 15)

Address(es): RSCAN.RMDF00 000A 83A8h, RSCAN.RMDF01 000A 83B8h, RSCAN.RMDF02 000A 83C8h, RSCAN.RMDF03 000A 83D8h, RSCAN.RMDF04 000A 83E8h, RSCAN.RMDF05 000A 83F8h, RSCAN.RMDF06 000A 8408h, RSCAN.RMDF07 000A 8418h, RSCAN.RMDF08 000A 8428h, RSCAN.RMDF09 000A 8438h, RSCAN.RMDF010 000A 8448h, RSCAN.RMDF011 000A 8458h, RSCAN.RMDF012 000A 8468h, RSCAN.RMDF013 000A 8478h, RSCAN.RMDF014 000A 8488h, RSCAN.RMDF015 000A 8498h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDB1[7:0] | | | | | | | | RMDB0[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RMDB0[7:0] | Receive Buffer Data Byte 0 | Data in the message stored in the receive buffer can be read. | R |
| b15 to b8 | RMDB1[7:0] | Receive Buffer Data Byte 1 | | R |

When the RMPTRn.RMDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

## 36.2.31 Receive Buffer Register nCH (RMDF1n) (n = 0 to 15)

Address(es): RSCAN.RMDF10 000A 83AAh, RSCAN.RMDF11 000A 83BAh, RSCAN.RMDF12 000A 83CAh, RSCAN.RMDF13 000A 83DAh, RSCAN.RMDF14 000A 83EAh, RSCAN.RMDF15 000A 83FAh, RSCAN.RMDF16 000A 840Ah, RSCAN.RMDF17 000A 841Ah, RSCAN.RMDF18 000A 842Ah, RSCAN.RMDF19 000A 843Ah, RSCAN.RMDF110 000A 844Ah, RSCAN.RMDF111 000A 845Ah, RSCAN.RMDF112 000A 846Ah, RSCAN.RMDF113 000A 847Ah, RSCAN.RMDF114 000A 848Ah, RSCAN.RMDF115 000A 849Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDB3[7:0] | | | | | | | | RMDB2[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RMDB2[7:0] | Receive Buffer Data Byte 2 | Data in the message stored in the receive buffer can be read. | R |
| b15 to b8 | RMDB3[7:0] | Receive Buffer Data Byte 3 | | R |

When the RMPTRn.RMDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

## 36.2.32 Receive Buffer Register nDL (RMDF2n) (n = 0 to 15)

Address(es): RSCAN.RMDF20 000A 83ACh, RSCAN.RMDF21 000A 83BCh, RSCAN.RMDF22 000A 83CCh, RSCAN.RMDF23 000A 83DCh, RSCAN.RMDF24 000A 83ECh, RSCAN.RMDF25 000A 83FCh, RSCAN.RMDF26 000A 840Ch, RSCAN.RMDF27 000A 841Ch, RSCAN.RMDF28 000A 842Ch, RSCAN.RMDF29 000A 843Ch, RSCAN.RMDF210 000A 844Ch, RSCAN.RMDF211 000A 845Ch, RSCAN.RMDF212 000A 846Ch, RSCAN.RMDF213 000A 847Ch, RSCAN.RMDF214 000A 848Ch, RSCAN.RMDF215 000A 849Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDB5[7:0] | | | | | | | | RMDB4[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RMDB4[7:0] | Receive Buffer Data Byte 4 | Data in the message stored in the receive buffer can be read. | R |
| b15 to b8 | RMDB5[7:0] | Receive Buffer Data Byte 5 | | R |

When the RMPTRn.RMDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

## 36.2.33 Receive Buffer Register nDH (RMDF3n) (n = 0 to 15)

Address(es): RSCAN.RMDF30 000A 83AEh, RSCAN.RMDF31 000A 83BEh, RSCAN.RMDF32 000A 83CEh, RSCAN.RMDF33 000A 83DEh, RSCAN.RMDF34 000A 83EEh, RSCAN.RMDF35 000A 83FEh, RSCAN.RMDF36 000A 840Eh, RSCAN.RMDF37 000A 841Eh, RSCAN.RMDF38 000A 842Eh, RSCAN.RMDF39 000A 843Eh, RSCAN.RMDF310 000A 844Eh, RSCAN.RMDF311 000A 845Eh, RSCAN.RMDF312 000A 846Eh, RSCAN.RMDF313 000A 847Eh, RSCAN.RMDF314 000A 848Eh, RSCAN.RMDF315 000A 849Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDB7[7:0] | | | | | | | | RMDB6[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RMDB6[7:0] | Receive Buffer Data Byte 6 | Data in the message stored in the receive buffer can be read. | R |
| b15 to b8 | RMDB7[7:0] | Receive Buffer Data Byte 7 | | R |

When the RMPTRn.RMDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

## 36.2.34 Receive FIFO Control Register m (RFCCm) (m = 0, 1)

Address(es): RSCAN.RFCC0 000A 8338h, RSCAN.RFCC1 000A 833Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFIGCV[2:0] | | | RFIM | — | RFDC[2:0] | | | — | — | — | — | — | — | RFIE | RFE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RFE | Receive FIFO Buffer Enable | 0: No receive FIFO buffer is used.<br>1: Receive FIFO buffers are used. | R/W |
| b1 | RFIE | Receive FIFO Interrupt Enable | 0: Receive FIFO interrupt is disabled.<br>1: Receive FIFO interrupt is enabled. | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b10 to b8 | RFDC[2:0] | Receive FIFO Buffer Depth Configuration | b10 b8<br>0 0 0: 0 messages<br>0 0 1: 4 messages<br>0 1 0: 8 messages<br>0 1 1: 16 messages<br>1 0 0: Setting prohibited<br>1 0 1: Setting prohibited<br>1 1 0: Setting prohibited<br>1 1 1: Setting prohibited | R/W |
| b11 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b12 | RFIM | Receive FIFO Interrupt Source Select | 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met.<br>1: An interrupt occurs each time a message has been received. | R/W |
| b15 to b13 | RFIGCV[2:0] | Receive FIFO Interrupt Request Timing Select | b15 b13<br>0 0 0: When FIFO is 1/8 full.<br>0 0 1: When FIFO is 2/8 full.<br>0 1 0: When FIFO is 3/8 full.<br>0 1 1: When FIFO is 4/8 full.<br>1 0 0: When FIFO is 5/8 full.<br>1 0 1: When FIFO is 6/8 full.<br>1 1 0: When FIFO is 7/8 full.<br>1 1 1: When FIFO is full. | R/W |

### RFE Bit (Receive FIFO Buffer Enable)

Setting the RFE bit to 1 makes receive FIFO buffers available. Setting this bit to 0 sets the RFSTSm.RFEMP flag to 1 (the receive FIFO buffer contains no unread message (buffer empty)). Modify this bit only in global operating mode or global test mode.

### RFIE Bit (Receive FIFO Interrupt Enable)

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit is set to 0 (no receive FIFO buffer is used).

### RFDC[2:0] Bits (Receive FIFO Buffer Depth Configuration)

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. If these bits are set to 000b, do not use any receive FIFO buffer. Modify these bits only in global reset mode.

### RFIM Bit (Receive FIFO Interrupt Source Select)

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

### RFIGCV[2:0] Bits (Receive FIFO Interrupt Request Timing Select)

These bits are used to specify the fraction of the transmit/receive FIFO buffer (the number of messages is selected by the setting of the RFDC[2:0] bits) that must be filled for the FIFO buffer to generate a receive interrupt request when the RFIM bit is set to 0.

When the RFDC[2:0] bits are set to 001b (4 messages), set the RFIGCV[2:0] bits to 001b, 011b, 101b, or 111b. Modify these bits only in global reset mode.

## 36.2.35 Receive FIFO Status Register m (RFSTSm) (m = 0, 1)

Address(es): RSCAN.RFSTS0 000A 8340h, RSCAN.RFSTS1 000A 8342h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | RFMC[5:0] | | | | | | — | — | — | — | RFIF | RFMLT | RFFLL | RFEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RFEMP | Receive FIFO Buffer Empty Status Flag | 0: The receive FIFO buffer contains unread messages.<br>1: The receive FIFO buffer contains no unread message (buffer empty). | R |
| b1 | RFFLL | Receive FIFO Buffer Full Status Flag | 0: The receive FIFO buffer is not full.<br>1: The receive FIFO buffer is full. | R |
| b2 | RFMLT | Receive FIFO Message Lost Flag | 0: No receive FIFO message is lost.<br>1: A receive FIFO message is lost. | R/(W) *1 |
| b3 | RFIF | Receive FIFO Interrupt Request Flag | 0: No receive FIFO interrupt request is present.<br>1: A receive FIFO interrupt request is present. | R/(W) *1 |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b13 to b8 | RFMC[5:0] | Receive FIFO Unread Message Counter | The number of unread messages stored in the receive FIFO buffer is displayed. | R |
| b15, b14 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

### RFEMP Flag (Receive FIFO Buffer Empty Status Flag)

This flag becomes 1 when all messages in the receive FIFO buffer have been read. This flag also becomes 1 when the RFCCm.RFE bit is 0 or in global reset mode.

This flag becomes 0 when even a single received message has been stored in the receive FIFO buffer.

### RFFLL Flag (Receive FIFO Buffer Full Status Flag)

This flag becomes 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFCCm.RFDC[2:0] bits.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFCCm.RFDC[2:0] bits, this flag becomes 0. This flag also becomes 0 when the RFCCm.RFE bit is set to 0 (no receive FIFO buffer is used) or in global reset mode.

### RFMLT Flag (Receive FIFO Message Lost Flag)

This flag becomes 1 when it is attempted to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag becomes 0 in global reset mode or by writing 0 to this flag.

Modify this bit only in global operating mode or global test mode.

**RFIF Flag (Receive FIFO Interrupt Request Flag)**

This flag becomes 1 when the receive FIFO interrupt request generation conditions set by the RFCCm.RFIGCV[2:0] bits (m = 0, 1) and the RFCCm.RFIM bit are met. This flag becomes 0 in global reset mode or by writing 0 to this flag. Modify this bit only in global operating mode or global test mode.

**RFMC[5:0] Flags (Receive FIFO Unread Message Counter)**

These flags indicate the number of unread messages in the receive FIFO buffer. This flag becomes 00h when the RFCCm.RFE bit is set to 0.

## 36.2.36 Receive FIFO Pointer Control Register m (RFPCTRm) (m = 0, 1)

Address(es): RSCAN.RFPCTR0 000A 8348h, RSCAN.RFPCTR1 000A 834Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | RFPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RFPC[7:0] | Receive FIFO Pointer | When these bits are set to FFh, the read pointer moves to the next unread message in the receive FIFO buffer. The setting for these bits must be FFh. | W |
| b15 to b8 | — | Reserved | The write value should be 0. | W |

**RFPC[7:0] Bits (Receive FIFO Pointer)**

When the RFPC[7:0] bits are set to FFh, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFSTSm.RFMC[5:0] (receive FIFO unread message counter) value is decremented. Read the RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m registers to read messages in the receive FIFO buffer, and then write FFh to the RFPC[7:0] bits.

Write FFh to these bits when the RFCCm.RFE bit is set to 1 (receive FIFO buffers are used) and the RFSTSm.RFEMP flag is 0 (the receive FIFO buffer contains unread messages).

## 36.2.37 Receive FIFO Access Register mAL (RFIDLm) (m = 0, 1)

Address(es): RSCAN.RFIDL0 000A 85A0h, RSCAN.RFIDL1 000A 85B0h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFID[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | RFID[15:0] | Receive FIFO Buffer ID Data L | The standard ID or extended ID of received message can be read.<br>Read bits 10 to 0 for standard ID. Bits 15 to 11 are read as 0. | R |

This register can be read when the GRWCR.RPAGE bit is 1.

**RFID[15:0] Bits (Receive FIFO Buffer ID Data L)**

These bits indicate the ID of the message stored in the receive FIFO buffer.

## 36.2.38 Receive FIFO Access Register mAH (RFIDHm) (m = 0, 1)

Address(es): RSCAN.RFIDH0 000A 85A2h, RSCAN.RFIDH1 000A 85B2h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFIDE | RFRTR | — | RFID[28:16] | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b12 to b0 | RFID[28:16] | Receive FIFO Buffer ID Data H | The standard ID or extended ID of received message can be read.<br>For standard ID, these bits are read as 0. | R |
| b13 | — | Reserved | This bit is read as 0. | R |
| b14 | RFRTR | Receive FIFO Buffer RTR | 0: Data frame<br>1: Remote frame | R |
| b15 | RFIDE | Receive FIFO Buffer IDE | 0: Standard ID<br>1: Extended ID | R |

This register can be read when the GRWCR.RPAGE bit is 1.

**RFID[28:16] Bits (Receive FIFO Buffer ID Data H)**

These bits indicate the ID of the message stored in the receive FIFO buffer.

**RFRTR Bit (Receive FIFO Buffer RTR)**

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

**RFIDE Bit (Receive FIFO Buffer IDE)**

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

## 36.2.39 Receive FIFO Access Register mBL (RFTSm) (m = 0, 1)

Address(es): RSCAN.RFTS0 000A 85A4h, RSCAN.RFTS1 000A 85B4h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | RFTS[15:0] | Receive FIFO Buffer Timestamp Data | Timestamp value of the received message can be read. | R |

This register can be read when the GRWCR.RPAGE bit is 1.

**RFTS[15:0] Bits (Receive FIFO Buffer Timestamp Data)**

These bits indicate the timestamp value of the message stored in the receive FIFO buffer.

## 36.2.40 Receive FIFO Access Register mBH (RFPTRm) (m = 0, 1)

Address(es): RSCAN.RFPTR0 000A 85A6h, RSCAN.RFPTR1 000A 85B6h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDLC[3:0] | | | | RFPTR[11:0] | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b11 to b0 | RFPTR[11:0] | Receive FIFO Buffer Label Data | Label information of the received message can be read. | R |
| b15 to b12 | RFDLC[3:0] | Receive FIFO Buffer DLC Data | b15 b14 b13 b12<br>0 0 0 0: 0 data bytes<br>0 0 0 1: 1 data byte<br>0 0 1 0: 2 data bytes<br>0 0 1 1: 3 data bytes<br>0 1 0 0: 4 data bytes<br>0 1 0 1: 5 data bytes<br>0 1 1 0: 6 data bytes<br>0 1 1 1: 7 data bytes<br>1 x x x: 8 data bytes | R |

x: Don't care

This register can be read when the GRWCR.RPAGE bit is 1.

**RFPTR[11:0] Bits (Receive FIFO Buffer Label Data)**

These bits indicate the label information of the message stored in the receive FIFO buffer.

**RFDLC[3:0] Bits (Receive FIFO Buffer DLC Data)**

These bits indicate the data length of the message stored in the receive FIFO buffer.

## 36.2.41 Receive FIFO Access Register mCL (RFDF0m) (m = 0, 1)

Address(es): RSCAN.RFDF00 000A 85A8h, RSCAN.RFDF01 000A 85B8h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDB1[7:0] | | | | | | | | RFDB0[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RFDB0[7:0] | Receive FIFO Buffer Data Byte 0 | Data in the message stored in the receive FIFO buffer can be read. | R |
| b15 to b8 | RFDB1[7:0] | Receive FIFO Buffer Data Byte 1 | | R |

When the RFPTRm.RFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

## 36.2.42 Receive FIFO Access Register mCH (RFDF1m) (m = 0, 1)

Address(es): RSCAN.RFDF10 000A 85AAh, RSCAN.RFDF11 000A 85BAh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDB3[7:0] | | | | | | | | RFDB2[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RFDB2[7:0] | Receive FIFO Buffer Data Byte 2 | Data in the message stored in the receive FIFO buffer can be read. | R |
| b15 to b8 | RFDB3[7:0] | Receive FIFO Buffer Data Byte 3 | | R |

When the RFPTRm.RFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

## 36.2.43 Receive FIFO Access Register mDL (RFDF2m) (m = 0, 1)

Address(es): RSCAN.RFDF20 000A 85ACh, RSCAN.RFDF21 000A 85BCh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDB5[7:0] | | | | | | | | RFDB4[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RFDB4[7:0] | Receive FIFO Buffer Data Byte 4 | Data in the message stored in the receive FIFO buffer can be read. | R |
| b15 to b8 | RFDB5[7:0] | Receive FIFO Buffer Data Byte 5 | | R |

When the RFPTRm.RFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

## 36.2.44 Receive FIFO Access Register mDH (RFDF3m) (m = 0, 1)

Address(es): RSCAN.RFDF30 000A 85AEh, RSCAN.RFDF31 000A 85BEh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDB7[7:0] | | | | | | | | RFDB6[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | RFDB6[7:0] | Receive FIFO Buffer Data Byte 6 | Data in the message stored in the receive FIFO buffer can be read. | R |
| b15 to b8 | RFDB7[7:0] | Receive FIFO Buffer Data Byte 7 | | R |

When the RFPTRm.RFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read when the GRWCR.RPAGE bit is 1.

## 36.2.45 Transmit/Receive FIFO Control Register 0L (CFCCL0)

Address(es): RSCAN0.CFCCL0 000A 8350h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFIGCV[2:0] | | | CFIM | — | CFDC[2:0] | | | — | — | — | — | — | CFTXIE | CFRXIE | CFE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CFE | Transmit/Receive FIFO Buffer Enable | 0: No transmit/receive FIFO buffer is used.<br>1: Transmit/receive FIFO buffers are used. | R/W |
| b1 | CFRXIE | Transmit/Receive FIFO Receive Interrupt Enable | 0: Transmit/receive FIFO receive interrupt is disabled.<br>1: Transmit/receive FIFO receive interrupt is enabled. | R/W |
| b2 | CFTXIE | Transmit/Receive FIFO Transmit Interrupt Enable | 0: Transmit/receive FIFO transmit interrupt is disabled.<br>1: Transmit/receive FIFO transmit interrupt is enabled. | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b10 to b8 | CFDC[2:0] | Transmit/Receive FIFO Buffer Depth Configuration | b10 b8<br>0 0 0: 0 messages<br>0 0 1: 4 messages<br>0 1 0: 8 messages<br>0 1 1: 16 messages<br>1 0 0: Setting prohibited<br>1 0 1: Setting prohibited<br>1 1 0: Setting prohibited<br>1 1 1: Setting prohibited | R/W |
| b11 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b12 | CFIM | Transmit/Receive FIFO Interrupt Source Select | 0:<br>• Receive mode<br>When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated.<br>• Transmit mode<br>When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated.<br>1:<br>• Receive mode<br>A FIFO receive interrupt request is generated each time a message has been received.<br>• Transmit mode<br>A FIFO transmit interrupt request is generated each time a message has been transmitted. | R/W |
| b15 to b13 | CFIGCV[2:0] | Transmit/Receive FIFO Receive Interrupt Request Timing Select | b15 b13<br>0 0 0: When FIFO is 1/8 full.<br>0 0 1: When FIFO is 2/8 full.<br>0 1 0: When FIFO is 3/8 full.<br>0 1 1: When FIFO is 4/8 full.<br>1 0 0: When FIFO is 5/8 full.<br>1 0 1: When FIFO is 6/8 full.<br>1 1 0: When FIFO is 7/8 full.<br>1 1 1: When FIFO is full. | R/W |

### CFE Bit (Transmit/Receive FIFO Buffer Enable)

Setting this bit to 1 makes transmit/receive FIFO buffers available.

When this bit is set to 0 in transmit mode, if a message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission, CAN bus error detection, or arbitration lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is set to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode: Channel reset mode

Modify this bit only in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode: Channel communication mode or channel halt mode

#### CFRXIE Bit (Transmit/Receive FIFO Receive Interrupt Enable)

When the CFSTS0.CFRXIF flag becomes 1 while this bit is 1, a transmit/receive FIFO receive interrupt request is generated.
Modify this bit with the CFE bit set to 0.

#### CFTXIE Bit (Transmit/Receive FIFO Transmit Interrupt Enable)

When the CFSTS0.CFTXIF flag becomes 1 while this bit is 1, a transmit/receive FIFO transmit interrupt request is generated.
Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

#### CFDC[2:0] Bits (Transmit/Receive FIFO Buffer Depth Configuration)

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer.
If these bits are set to 000b, do not use any receive FIFO buffer. Modify these bits only in global reset mode.

#### CFIM Bit (Transmit/Receive FIFO Interrupt Source Select)

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

#### CFIGCV[2:0] Bits (Transmit/Receive FIFO Receive Interrupt Request Timing Select)

These bits are used to specify the fraction of the transmit/receive FIFO buffer (the number of messages is selected by the setting of the CFDC[2:0] bits) that must be filled for the FIFO buffer to generate a receive interrupt request when the CFCCH0.CFM[1:0] bits are set to 00b (receive mode) and the CFIM bit is set to 0.
When the CFDC[2:0] bits are set to 001b (4 messages), set the CFIGCV[2:0] bits to 001b, 011b, 101b, or 111b.
Modify these bits only in global reset mode.

## 36.2.46 Transmit/Receive FIFO Control Register 0H (CFCCH0)

Address(es): RSCAN0.CFCCH0 000A 8352h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFITT[7:0] | | | | | | | | — | — | CFTML[1:0] | | CFITR | CFITSS | CFM[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CFM[1:0] | Transmit/Receive FIFO Mode Select | b1 b0<br>0 0: Receive mode<br>0 1: Transmit mode<br>1 0: Setting prohibited<br>1 1: Setting prohibited | R/W |
| b2 | CFITSS | Interval Timer Clock Source Select | 0: Clock selected by the CFITR bit<br>1: CAN bit time clock | R/W |
| b3 | CFITR | Transmit/Receive FIFO Interval Timer Resolution | 0: Clock obtained by frequency-dividing PCLK by the ITRCP[15:0] value<br>1: Clock obtained by frequency-dividing PCLK by the ITRCP[15:0] value × 10 | R/W |
| b5, b4 | CFTML[1:0] | Transmit Buffer Link Configuration | Set the transmit buffer number to be linked to the transmit/receive FIFO buffer. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b8 | CFITT[7:0] | Message Transmission Interval Configuration | Set a message transmission interval.<br>Set these bits to a value within a range of 00h to FFh. | R/W |

### CFM[1:0] Bits (Transmit/Receive FIFO Mode Select)

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

### CFITSS Bit (Interval Timer Clock Source Select)

Setting this bit to 0 selects the clock selected by the CFITR bit as the clock source for counting by the interval timer.
Setting this bit to 1 selects the CAN bit time clock as the clock source for counting by the interval timer. Set the CFCCL0.CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITSS bit.

### CFITR Bit (Transmit/Receive FIFO Interval Timer Resolution)

This bit is valid when the setting of the CFITSS bit is 0.
Setting this bit to 0 selects the clock obtained by frequency-dividing PCLK by the GCFGH.ITRCP[15:0] value.
Setting this bit to 1 selects the clock obtained by frequency-dividing PCLK by the GCFGH.ITRCP[15:0] value × 10.
Modifying this bit with the CFCCL0.CFE bit set to 0 (no transmit/receive FIFO buffer is used).

### CFTML[1:0] Bits (Transmit Buffer Link Configuration)

These bits are used to set the number of transmit buffer to be linked to the transmit/receive FIFO buffer when the CFM[1:0] bits are set to 01b (transmit mode).
Setting the CFCCL0.CFDC[2:0] bits to 001b or more enables the setting of the CFTML[1:0] bits.
Modify these bits only in global reset mode.

### CFITT[7:0] Bits (Message Transmission Interval Configuration)

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01b (transmit mode).
Set the CFCCL0.CFE bit to 0 (no transmit/receive FIFO buffer is used) and then modify the CFITT[7:0] bits.

## 36.2.47 Transmit/Receive FIFO Status Register 0 (CFSTS0)

Address(es): RSCAN0.CFSTS0 000A 8358h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | CFMC[5:0] | | | | | | — | — | — | CFTXIF | CFRXIF | CFMLT | CFFLL | CFEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CFEMP | Transmit/Receive FIFO Buffer Empty Status Flag | 0: The transmit/receive FIFO buffer contains messages.<br>1: The transmit/receive FIFO buffer contains no message (buffer empty). | R |
| b1 | CFFLL | Transmit/Receive FIFO Buffer Full Status Flag | 0: The transmit/receive FIFO buffer is not full.<br>1: The transmit/receive FIFO buffer is full. | R |
| b2 | CFMLT | Transmit/Receive FIFO Message Lost Flag | 0: No transmit/receive FIFO message is lost.<br>1: A transmit/receive FIFO message is lost. | R/(W) *1 |
| b3 | CFRXIF | Transmit/Receive FIFO Receive Interrupt Request Flag | 0: No transmit/receive FIFO receive interrupt request is present.<br>1: A transmit/receive FIFO receive interrupt request is present. | R/(W) *1 |
| b4 | CFTXIF | Transmit/Receive FIFO Transmit Interrupt Request Flag | 0: No transmit/receive FIFO transmit interrupt request is present.<br>1: A transmit/receive FIFO transmit interrupt request is present. | R/(W) *1 |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b13 to b8 | CFMC[5:0] | Transmit/Receive FIFO Message Counter | The number of messages stored in the transmit/receive FIFO buffer is indicated. | R |
| b15, b14 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

### CFEMP Flag (Transmit/Receive FIFO Buffer Empty Status Flag)

[Setting conditions]

- When the CFCCH0.CFM[1:0] value is 00b: All messages have been read, or global reset mode.
- When the CFCCH0.CFM[1:0] value is 01b: All messages have been transmitted, or channel reset mode.
- When the CFCCL0.CFE value is 0 (no transmit/receive FIFO buffer is used).
  Note that this flag becomes 1 after transmission completion, CAN bus error detection, or arbitration lost when the message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next.

[Clearing conditions]

- When the CFCCH0.CFM[1:0] value is 00b: Any one of received messages has been stored in the transmit/receive FIFO buffer.
- When the CFCCH0.CFM[1:0] value is 01b: A value FFh has been written to the CFPCTR0 register after data was written to the CFIDL0, CFIDH0, CFPTR0, and CFDF00 to CFDF30 registers.

### CFFLL Flag (Transmit/Receive FIFO Buffer Full Status Flag)

[Setting condition]

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFCCL0.CFDC[2:0] bits.

[Clearing conditions]

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFCCL0.CFDC[2:0] bits.

- When the CFCCL0.CFE value is 0 (no transmit/receive FIFO buffer is used).
  Note that this flag becomes 0 after transmission completion, CAN bus error detection, or arbitration lost when the message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next.
- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

#### CFMLT Flag (Transmit/Receive FIFO Message Lost Flag)

[Setting condition]

- When it is attempted to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

[Clearing conditions]

- Write 0 to the CFMLT flag
- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

Set this flag to 0 in global operating mode or global test mode.

#### CFRXIF Flag (Transmit/Receive FIFO Receive Interrupt Request Flag)

[Setting condition]

- When CFCCH0.CFM[1:0] value is 00b and interrupt source setting the CFCCL0.CFIM bit is generated.

[Clearing conditions]

- Write 0 to the CFRXIF flag
- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

Set this flag to 0 in global operating mode or global test mode.

#### CFTXIF Flag (Transmit/Receive FIFO Transmit Interrupt Request Flag)

[Setting condition]

- When CFCCH0.CFM[1:0] value is 01b and interrupt source setting the CFCCL0.CFIM bit is generated.

[Clearing conditions]

- Write 0 to the CFTXIF flag
- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

Set this flag to 0 in global operating mode or global test mode.

#### CFMC[5:0] Flags (Transmit/Receive FIFO Message Counter)

The CFMC[5:0] flags indicate the following values that depend on the setting of the CFCCH0.CFM[1:0] bits.

- When CFM[1:0] value is 01b (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00b (receive mode): Number of unread received messages in the buffer

These bits are set to 0 when any of the following conditions is met.

- When CFCCH0.CFM[1:0] value is 00b: In global reset mode
- When CFCCH0.CFM[1:0] value is 01b: In channel reset mode

## 36.2.48 Transmit/Receive FIFO Pointer Control Register 0 (CFPCTR0)

Address(es): RSCAN0.CFPCTR0 000A 835Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CFPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | CFPC[7:0] | RSCAN0 Transmit/Receive FIFO Pointer | Receive mode:<br>Writing FFh to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer.<br>Transmit mode:<br>Writing FFh to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. | W |
| b15 to b8 | — | Reserved | The write value should be 0. | W |

### CFPC[7:0] Bits (RSCAN0 Transmit/Receive FIFO Pointer)

Receive mode (CFCCH0.CFM[1:0] value is 00b):
Writing FFh to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFSTS0.CFMC[5:0] value (transmit/receive FIFO message counter) is decremented. Read the CFIDL0, CFIDH0, CFTS0, CFPTR0, and CFDF00 to CFDF30 registers to read messages in the transmit/receive FIFO buffer, and then write FFh to the CFPC[7:0] bits.
Write FFh to these bits when the CFCCL0.CFE bit is 1 (transmit/receive FIFO buffers are used) and the CFSTS0.CFEMP flag is 0 (the transmit/receive FIFO buffer contains messages).

Transmit mode (CFCCH0.CFM[1:0] value is 01b):
Writing FFh to the CFPC[7:0] bits stores the data written to the CFIDL0, CFIDH0, CFPTR0, and CFDF00 to CFDF30 registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFSTS0.CFMC[5:0] value is incremented. Write transmit messages to the CFIDL0, CFIDH0, CFPTR0, and CFDF00 to CFDF30 registers and then write FFh to the CFPC[7:0] bits.
Write FFh to these bits when the CFCCL0.CFE bit is 1 and the CFSTS0.CFFLL flag is 0 (the transmit/receive FIFO buffer is not full).

## 36.2.49 Transmit/Receive FIFO Access Register 0AL (CFIDL0)

Address(es): RSCAN0.CFIDL0 000A 85E0h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFID[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | CFID[15:0] | Transmit/Receive FIFO Buffer ID Data L | When CFCCH0.CFM[1:0] value is 01b (transmit mode):<br>Set standard ID or extended ID. For standard ID, write an ID to b10 to b0 and write 0 to b15 to b11.<br>When CFCCH0.CFM[1:0] value is 00b (receive mode):<br>Standard ID or extended ID in the received message can be read. For standard ID, read b10 to b0. The value read from b15 to b11 are 0. | R/W |

Modify this register only when the CFCCH0.CFM[1:0] value is 01b (transmit mode). This register is readable only when the CFCCH0.CFM[1:0] value is 00b (receive mode).
This register can be read/written when the GRWCR.RPAGE bit is 1.

**CFID[15:0] Bits (Transmit/Receive FIFO Buffer ID Data L)**

These bits indicate the ID of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b.
When the CFCCH0.CFM[1:0] value is 01b, set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

## 36.2.50 Transmit/Receive FIFO Access Register 0AH (CFIDH0)

Address(es): RSCAN0.CFIDH0 000A 85E2h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFIDE | CFRTR | THLEN | CFID[28:16] | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b12 to b0 | CFID[28:16] | Transmit/Receive FIFO Buffer ID Data H | When CFCCH0.CFM[1:0] value is 01b (transmit mode):<br>Set standard ID or extended ID. For standard ID, write 0 to these bits.<br>When CFCCH0.CFM[1:0] value is 00b (receive mode):<br>Standard ID or extended ID in the received message can be read. For standard ID, these bits are read as 0. | R/W |
| b13 | THLEN | Transmit History Data Store Enable | This bit is valid only when the CFCCH0.CFM[1:0] value is 01b (transmit mode).<br>0: Transmit history data is not stored in the buffer.<br>1: Transmit history data is stored in the buffer. | R/W |
| b14 | CFRTR | Transmit/Receive FIFO Buffer RTR | 0: Data frame<br>1: Remote frame | R/W |
| b15 | CFIDE | Transmit/Receive FIFO Buffer IDE | 0: Standard ID<br>1: Extended ID | R/W |

Modify this register only when the CFCCH0.CFM[1:0] value is 01b (transmit mode).
This register is readable only when the CFCCH0.CFM[1:0] value is 00b (receive mode).
This register can be read/written when the GRWCR.RPAGE bit is 1.

### CFID[28:16] Bits (Transmit/Receive FIFO Buffer ID Data H)

These bits indicate the ID of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b.
When the CFCCH0.CFM[1:0] value is 01b, set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

### THLEN Bit (Transmit History Data Store Enable)

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.
This bit is enabled when the CFCCH0.CFM[1:0] value is 01b (transmit mode).

### CFRTR Bit (Transmit/Receive FIFO Buffer RTR)

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b. When the CFCCH0.CFM[1:0] value is 01b, set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

### CFIDE Bit (Transmit/Receive FIFO Buffer IDE)

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b. When the CFCCH0.CFM[1:0] value is 01b, set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

## 36.2.51 Transmit/Receive FIFO Access Register 0BL (CFTS0)

Address(es): RSCAN0.CFTS0 000A 85E4h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | CFTS[15:0] | Transmit/Receive FIFO Buffer Timestamp Data | These bits are valid only when the CFCCH0.CFM[1:0] value is 00b (receive mode).<br>The timestamp value of the received message can be read. | R |

This register can be read when the GRWCR.RPAGE bit is 1.

**CFTS[15:0] Bits (Transmit/Receive FIFO Buffer Timestamp Data)**

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFCCH0.CFM[1:0] value is 00b.

## 36.2.52 Transmit/Receive FIFO Access Register 0BH (CFPTR0)

Address(es): RSCAN0.CFPTR0 000A 85E6h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDLC[3:0] | | | | CFPTR[11:0] | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b11 to b0 | CFPTR[11:0] | Transmit/Receive FIFO Buffer Label Data | When CFCCH0.CFM[1:0] value is 01b (transmit mode): Set the label information to be stored in the transmit history buffer. Only CFPTR[7:0] bits are valid.<br>When CFCCH0.CFM[1:0] value is 00b (receive mode): The label information of the received message can be read. | R/W |
| b15 to b12 | CFDLC[3:0] | Transmit/Receive FIFO Buffer DLC Data | b15 b12<br>0 0 0 0: 0 data bytes<br>0 0 0 1: 1 data byte<br>0 0 1 0: 2 data bytes<br>0 0 1 1: 3 data bytes<br>0 1 0 0: 4 data bytes<br>0 1 0 1: 5 data bytes<br>0 1 1 0: 6 data bytes<br>0 1 1 1: 7 data bytes<br>1 x x x: 8 data bytes | R/W |

x: Don’t care

Modify this register only when the CFCCH0.CFM[1:0] value is 01b (transmit mode).
This register is readable only when the CFCCH0.CFM[1:0] value is 00b (receive mode).
This register can be read/written when the GRWCR.RPAGE bit is 1.

### CFPTR[11:0] Bits (Transmit/Receive FIFO Buffer Label Data)

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b. When the CFCCH0.CFM[1:0] value is 01b, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

### CFDLC[3:0] Bits (Transmit/Receive FIFO Buffer DLC Data)

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFCCH0.CFM[1:0] value is 00b. When the CFCCH0.CFM[1:0] value is 01b, set the data length of the message to be transmitted from the transmit/receive FIFO buffer.
If 9-byte or more data length is set, 8 bytes of data is actually transmitted.

## 36.2.53 Transmit/Receive FIFO Access Register 0CL (CFDF00)

Address(es): RSCAN0.CFDF00 000A 85E8h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDB1[7:0] | | | | | | | | CFDB0[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | CFDB0[7:0] | Transmit/Receive FIFO Buffer Data Byte 0 | When CFCCH0.CFM[1:0] value is 01b (transmit mode):<br>Set the transmit/receive FIFO buffer data.<br>When CFCCH0.CFM[1:0] value is 00b (receive mode):<br>The message data stored in the transmit/receive FIFO buffer can be read. | R/W |
| b15 to b8 | CFDB1[7:0] | Transmit/Receive FIFO Buffer Data Byte 1 | | R/W |

Modify this register only when the CFCCH0.CFM[1:0] value is 01b.
This register is readable only when the CFCCH0.CFM[1:0] value is 00b. When the CFPTR0.CFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read/written when the GRWCR.RPAGE bit is 1.

## 36.2.54 Transmit/Receive FIFO Access Register 0CH (CFDF10)

Address(es): RSCAN0.CFDF10 000A 85EAh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDB3[7:0] | | | | | | | | CFDB2[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | CFDB2[7:0] | Transmit/Receive FIFO Buffer Data Byte 2 | When CFCCH0.CFM[1:0] value is 01b (transmit mode):<br>Set the transmit/receive FIFO buffer data.<br>When CFCCH0.CFM[1:0] value is 00b (receive mode):<br>The message data stored in the transmit/receive FIFO buffer can be read. | R/W |
| b15 to b8 | CFDB3[7:0] | Transmit/Receive FIFO Buffer Data Byte 3 | | R/W |

Modify this register only when the CFCCH0.CFM[1:0] value is 01b.
This register is readable only when the CFCCH0.CFM[1:0] value is 00b. When the CFPTR0.CFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read/written when the GRWCR.RPAGE bit is 1.

## 36.2.55 Transmit/Receive FIFO Access Register 0DL (CFDF20)

Address(es): RSCAN0.CFDF20 000A 85ECh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDB5[7:0] | | | | | | | | CFDB4[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | CFDB4[7:0] | Transmit/Receive FIFO Buffer Data Byte 4 | When CFCCH0.CFM[1:0] value is 01b (transmit mode):<br>Set the transmit/receive FIFO buffer data.<br>When CFCCH0.CFM[1:0] value is 00b (receive mode):<br>The message data stored in the transmit/receive FIFO buffer can be read. | R/W |
| b15 to b8 | CFDB5[7:0] | Transmit/Receive FIFO Buffer Data Byte 5 | | R/W |

Modify this register only when the CFCCH0.CFM[1:0] value is 01b.
This register is readable only when the CFCCH0.CFM[1:0] value is 00b. When the CFPTR0.CFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read/written when the GRWCR.RPAGE bit is 1.

## 36.2.56 Transmit/Receive FIFO Access Register 0DH (CFDF30)

Address(es): RSCAN0.CFDF30 000A 85EEh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDB7[7:0] | | | | | | | | CFDB6[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | CFDB6[7:0] | Transmit/Receive FIFO Buffer Data Byte 6 | When CFCCH0.CFM[1:0] value is 01b (transmit mode):<br>Set the transmit/receive FIFO buffer data.<br>When CFCCH0.CFM[1:0] value is 00b (receive mode):<br>The message data stored in the transmit/receive FIFO buffer can be read. | R/W |
| b15 to b8 | CFDB7[7:0] | Transmit/Receive FIFO Buffer Data Byte 7 | | R/W |

Modify this register only when the CFCCH0.CFM[1:0] value is 01b.
This register is readable only when the CFCCH0.CFM[1:0] value is 00b. When the CFPTR0.CFDLC[3:0] value is smaller than 1000b, data bytes for which no data is set are read as 00h.
This register can be read/written when the GRWCR.RPAGE bit is 1.

## 36.2.57 Receive FIFO Message Lost Status Register (RFMSTS)

Address(es): RSCAN.RFMSTS 000A 8360h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | RF1MLT | RF0MLT |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RF0MLT | Receive FIFO Buffer 0 Message Lost Status Flag | 0: No receive FIFO buffer m message is lost (m = 0, 1).<br>1: A receive FIFO buffer m message is lost. | R |
| b1 | RF1MLT | Receive FIFO Buffer 1 Message Lost Status Flag | | R |
| b7 to b2 | — | Reserved | These bits are read as 0. | R |

The RFMSTS register is set to 00h in global reset mode.

**RFmMLT Flag (Receive FIFO Buffer m Message Lost Status Flag)**

The RFmMLT flag becomes 1 when the RFSTSm.RFMLT flag becomes 1 (a receive FIFO message is lost). When the RFSTSm.RFMLT flag is set to 0, the RFmMLT flag becomes 0.

## 36.2.58 Transmit/Receive FIFO Message Lost Status Register (CFMSTS)

Address(es): RSCAN0.CFMSTS 000A 8361h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CF0MLT |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CF0MLT | RSCAN0 Transmit/Receive FIFO Buffer 0 Message Lost Status Flag | 0: No RSCAN0 transmit/receive FIFO buffer 0 message is lost.<br>1: An RSCAN0 transmit/receive FIFO buffer 0 message is lost. | R |
| b7 to b1 | — | Reserved | These bits are read as 0. | R |

The CFMSTS register is set to 00h in global reset mode.

**CF0MLT Flag (RSCAN0 Transmit/Receive FIFO Buffer 0 Message Lost Status Flag)**

The CF0MLT flag becomes 1 when the CFSTS0.CFMLT flag becomes 1 (a transmit/receive FIFO message is lost). When the CFSTS0.CFMLT flag is set to 0, the CF0MLT flag becomes 0.

## 36.2.59 Receive FIFO Interrupt Status Register (RFISTS)

Address(es): RSCAN.RFISTS 000A 8362h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | RF1IF | RF0IF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RF0IF | Receive FIFO Buffer 0 Interrupt Request Status Flag | 0: No receive FIFO buffer m interrupt request is present (m = 0, 1).<br>1: A receive FIFO buffer m interrupt request is present. | R |
| b1 | RF1IF | Receive FIFO Buffer 1 Interrupt Request Status Flag | | R |
| b7 to b2 | — | Reserved | These bits are read as 0. | R |

The RFISTS register is set to 00h in global reset mode.

**RFmIF Flag (Receive FIFO Buffer m Interrupt Request Status Flag)**

The RFmIF flag becomes 1 when the RFSTSm.RFIF flag becomes 1 (a receive FIFO interrupt request is present). When the RFSTSm.RFIF flag is set to 0, the RFmIF flag becomes 0.

## 36.2.60 Transmit/Receive FIFO Receive Interrupt Status Register (CFISTS)

Address(es): RSCAN.CFISTS 000A 8363h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CF0IF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CF0IF | RSCAN0 Transmit/Receive FIFO Buffer 0 Receive Interrupt Request Status Flag | 0: No RSCAN0 transmit/receive FIFO buffer 0 receive interrupt request is present.<br>1: An RSCAN0 transmit/receive FIFO buffer 0 receive interrupt request is present. | R |
| b7 to b1 | — | Reserved | These bits are read as 0. | R |

The CFISTS register is set to 00h in global reset mode.

**CF0IF Flag (RSCAN0 Transmit/Receive FIFO Buffer 0 Receive Interrupt Request Status Flag)**

The CF0IF flag becomes 1 when the CFSTS0.CFRXIF flag becomes 1 (a transmit/receive FIFO receive interrupt request is present). When the CFSTS0.CFRXIF flag is set to 0, the CF0IF flag becomes 0.

## 36.2.61 Transmit Buffer Control Register p (TMCp) (p = 0 to 3)

Address(es): RSCAN0.TMC0 000A 8364h, RSCAN0.TMC1 000A 8365h, RSCAN0.TMC2 000A 8366h, RSCAN0.TMC3 000A 8367h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TMOM | TMTAR | TMTR |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMTR | Transmit Request | 0: Transmission is not requested.<br>1: Transmission is requested. | R/(W) *1 |
| b1 | TMTAR | Transmit Abort Request | 0: Transmit abort is not requested.<br>1: Transmit abort is requested. | R/(W) *1 |
| b2 | TMOM | One-Shot Transmission Enable | 0: One-shot transmission is disabled.<br>1: One-shot transmission is enabled. | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 1 can be written to this bit. Writing 0 does not affect the bit value.

When the TMCp register meets the following condition, set it to 00h.

- The TMCp register corresponds to the transmit buffer number selected by the CFCCH0.CFTML[1:0] bits.

Bits in the TMCp register become all 0s in channel reset mode. Modify the TMCp register (p = 0 to 3) only in channel communication mode or channel halt mode.

### TMTR Bit (Transmit Request)

Setting this bit to 1 transmits the message stored in the transmit buffer.
The TMTR bit becomes 0 when any of the following conditions is met, but does not become 0 by writing 0 by the program.

- Transmission has been completed.
- Transmit abort has been completed by setting the TMTAR bit to 1.
- An error or arbitration lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the TMSTSp.TMTRF[1:0] value is 00b.

### TMTAR Bit (Transmit Abort Request)

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or to be transmitted next cannot be aborted.
When the TMTR bit becomes 1, the TMTAR bit can be set to 1.
The TMTAR bit is set to 0 when any of the following conditions is met, but does not become 0 by writing 0 by the program.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration lost has been detected.

If this bit becomes 0 at the timing when the program writes 1 to this bit, this bit becomes 0.

### TMOM Bit (One-Shot Transmission Enable)

Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMSTSp.TMTRM flag is 0. To set the TMOM bit to 1, also set the TMTR bit together.

## 36.2.62 Transmit Buffer Status Register p (TMSTSp) (p = 0 to 3)

Address(es): RSCAN0.TMSTS0 000A 836Ch, RSCAN0.TMSTS1 000A 836Dh, RSCAN0.TMSTS2 000A 836Eh, RSCAN0.TMSTS3 000A 836Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | TMTARM | TMTRM | TMTRF[1:0] | | TMTSTS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMTSTS | Transmit Buffer Transmit Status Flag | 0: Transmission is not in progress.<br>1: Transmission is in progress. | R |
| b2, b1 | TMTRF[1:0] | Transmit Buffer Transmit Result Flag | b2 b1<br>0 0: Transmission is in progress or no transmit request is present.<br>0 1: Transmit abort has been completed.<br>1 0: Transmission has been completed (without transmit abort request).<br>1 1: Transmission has been completed (with transmit abort request). | R/W |
| b3 | TMTRM | Transmit Buffer Transmit Request Status Flag | 0: No transmit request is present.<br>1: A transmit request is present. | R |
| b4 | TMTARM | Transmit Buffer Transmit Abort Request Status Flag | 0: No transmit abort request is present.<br>1: A transmit abort request is present. | R |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R |

The TMSTSp register becomes all 0s in channel reset mode.

### TMTSTS Flag (Transmit Buffer Transmit Status Flag)

This flag becomes 1 when transmission from the transmit buffer starts, and becomes 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

### TMTRF[1:0] Flags (Transmit Buffer Transmit Result Flag)

These flags indicate the result of transmission from the transmit buffer.
00b: Transmission is in progress or no transmit request is present.
01b: Transmission from the transmit buffer was aborted.
10b: Transmission has been completed with the TMCp.TMTAR bit set to 0 (transmit abort is not requested).
11b: Transmission has been completed with the TMCp.TMTAR bit set to 1 (transmit abort is requested).
Write 00b to the TMTRF[1:0] flags in channel communication mode or channel halt mode. Do not write any value other than 00b to these flags.

### TMTRM Flag (Transmit Buffer Transmit Request Status Flag)

The TMTRM flag becomes 1 when the TMCp.TMTR bit is set to 1, and becomes 0 when the TMCp.TMTR bit is set to 0.

### TMTARM Flag (Transmit Buffer Transmit Abort Request Status Flag)

The TMTARM flag becomes 1 when the TMCp.TMTAR bit is set to 1, and becomes 0 when the TMCp.TMTAR bit is set to 0.

## 36.2.63 Transmit Buffer Transmit Request Status Register (TMTRSTS)

Address(es): RSCAN0.TMTRSTS 000A 8374h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | TMTRSTS3 | TMTRSTS2 | TMTRSTS1 | TMTRSTS0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMTRSTS0 | RSCAN0 Transmit Buffer 0 Transmit Request Status Flag | 0: No transmit request is present.<br>1: A transmit request is present. | R |
| b1 | TMTRSTS1 | RSCAN0 Transmit Buffer 1 Transmit Request Status Flag | | R |
| b2 | TMTRSTS2 | RSCAN0 Transmit Buffer 2 Transmit Request Status Flag | | R |
| b3 | TMTRSTS3 | RSCAN0 Transmit Buffer 3 Transmit Request Status Flag | | R |
| b15 to b4 | — | Reserved | These bits are read as 0. | R |

**TMTRSTSp Flag (RSCAN0 Transmit Buffer p Transmit Request Status Flag) (p = 0 to 3)**

This flag indicates the status of the TMCp.TMTR bit.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag becomes 1.

The corresponding TMTRSTSp flag becomes 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

## 36.2.64 Transmit Buffer Transmit Complete Status Register (TMTCSTS)

Address(es): RSCAN0.TMTCSTS 000A 8376h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | TMTCSTS3 | TMTCSTS2 | TMTCSTS1 | TMTCSTS0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMTCSTS0 | RSCAN0 Transmit Buffer 0 Transmit Complete Status Flag | 0: Transmission has not been completed.<br>1: Transmission has been completed. | R |
| b1 | TMTCSTS1 | RSCAN0 Transmit Buffer 1 Transmit Complete Status Flag | | R |
| b2 | TMTCSTS2 | RSCAN0 Transmit Buffer 2 Transmit Complete Status Flag | | R |
| b3 | TMTCSTS3 | RSCAN0 Transmit Buffer 3 Transmit Complete Status Flag | | R |
| b15 to b4 | — | Reserved | These bits are read as 0. | R |

**TMTCSTSp Flag (RSCAN0 Transmit Buffer p Transmit Complete Status Flag) (p = 0 to 3)**

When the TMSTSp.TMTRF[1:0] flags become 10b (transmission has been completed (without transmit abort request)) or 11b (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag becomes 1. This flag becomes 0 when the corresponding TMSTSp.TMTRF[1:0] flags are set to 00b or in channel reset mode.

## 36.2.65 Transmit Buffer Transmit Abort Status Register (TMTASTS)

Address(es): RSCAN0.TMTASTS 000A 8378h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | TMTASTS3 | TMTASTS2 | TMTASTS1 | TMTASTS0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMTASTS0 | RSCAN0 Transmit Buffer 0 Transmit Abort Status Flag | 0: Transmission is not aborted.<br>1: Transmission is aborted. | R |
| b1 | TMTASTS1 | RSCAN0 Transmit Buffer 1 Transmit Abort Status Flag | | R |
| b2 | TMTASTS2 | RSCAN0 Transmit Buffer 2 Transmit Abort Status Flag | | R |
| b3 | TMTASTS3 | RSCAN0 Transmit Buffer 3 Transmit Abort Status Flag | | R |
| b15 to b4 | — | Reserved | These bits are read as 0. | R |

**TMTASTSp Flag (RSCAN0 Transmit Buffer p Transmit Abort Status Flag) (p = 0 to 3)**

When the TMSTSp.TMTRF[1:0] flags become 01b (transmit abort has been completed), the corresponding TMTASTSp flag becomes 1.

This flag becomes 0 when the corresponding TMSTSp.TMTRF[1:0] flags are set to 00b or in channel reset mode.

## 36.2.66 Transmit Buffer Interrupt Enable Register (TMIEC)

Address(es): RSCAN0.TMIEC 000A 837Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | TMIE3 | TMIE2 | TMIE1 | TMIE0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMIE0 | RSCAN0 Transmit Buffer 0 Interrupt Enable | 0: Transmit buffer interrupt is disabled.<br>1: Transmit buffer interrupt is enabled. | R/W |
| b1 | TMIE1 | RSCAN0 Transmit Buffer 1 Interrupt Enable | | R/W |
| b2 | TMIE2 | RSCAN0 Transmit Buffer 2 Interrupt Enable | | R/W |
| b3 | TMIE3 | RSCAN0 Transmit Buffer 3 Interrupt Enable | | R/W |
| b15 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

**TMIEp Bit (RSCAN0 Transmit Buffer p Interrupt Enable) (p = 0 to 3)**

When TMIEp bit is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify this bit when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers.

## 36.2.67 Transmit Buffer Register pAL (TMIDLp) (p = 0 to 3)

Address(es): RSCAN0.TMIDL0 000A 8600h, RSCAN0.TMIDL1 000A 8610h, RSCAN0.TMIDL2 000A 8620h, RSCAN0.TMIDL3 000A 8630h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMID[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | TMID[15:0] | Transmit Buffer ID Data L | Set standard ID or extended ID.<br>For standard ID, write an ID to b10 to b0 and write 0 to b15 to b11. | R/W |

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the GRWCR.RPAGE bit is 1.

**TMID[15:0] Bits (Transmit Buffer ID Data L)**

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

## 36.2.68 Transmit Buffer Register pAH (TMIDHp) (p = 0 to 3)

Address(es): RSCAN0.TMIDH0 000A 8602h, RSCAN0.TMIDH1 000A 8612h, RSCAN0.TMIDH2 000A 8622h, RSCAN0.TMIDH3 000A 8632h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMIDE | TMRTR | THLEN | TMID[28:16] | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b12 to b0 | TMID[28:16] | Transmit Buffer ID Data H | Set standard ID or extended ID.<br>For standard ID, write 0 to these bits. | R/W |
| b13 | THLEN | Transmit History Data Store Enable | 0: Transmit history data is not stored in the buffer.<br>1: Transmit history data is stored in the buffer. | R/W |
| b14 | TMRTR | Transmit Buffer RTR | 0: Data frame<br>1: Remote frame | R/W |
| b15 | TMIDE | Transmit Buffer IDE | 0: Standard ID<br>1: Extended ID | R/W |

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

### TMID[28:16] Bits (Transmit Buffer ID Data H)

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

### THLEN Bit (Transmit History Data Store Enable)

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

### TMRTR Bit (Transmit Buffer RTR)

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

### TMIDE Bit (Transmit Buffer IDE)

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

## 36.2.69 Transmit Buffer Register pBH (TMPTRp) (p = 0 to 3)

Address(es): RSCAN0.TMPTR0 000A 8606h, RSCAN0.TMPTR1 000A 8616h, RSCAN0.TMPTR2 000A 8626h, RSCAN0.TMPTR3 000A 8636h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDLC[3:0] | | | | — | — | — | — | TMPTR[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | TMPTR[7:0] | Transmit Buffer Label Data | Set the label information to be stored in the transmit history buffer. | R/W |
| b11 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b12 | TMDLC[3:0] | Transmit Buffer DLC Data | b15 b12<br>0 0 0 0: 0 data bytes<br>0 0 0 1: 1 data byte<br>0 0 1 0: 2 data bytes<br>0 0 1 1: 3 data bytes<br>0 1 0 0: 4 data bytes<br>0 1 0 1: 5 data bytes<br>0 1 1 0: 6 data bytes<br>0 1 1 1: 7 data bytes<br>1 x x x: 8 data bytes | R/W |

x: Don’t care

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

### TMPTR[7:0] Bits (Transmit Buffer Label Data)

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

### TMDLC[3:0] Bits (Transmit Buffer DLC Data)

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMIDHp.TMRTR bit is set to 0 (data frame). If a 9-byte (or more) data length is set, 8 bytes of data is actually transmitted.
When the TMIDHp.TMRTR bit is set to 1 (remote frame), set the data length of messages to be requested.

## 36.2.70 Transmit Buffer Register pCL (TMDF0p) (p = 0 to 3)

Address(es): RSCAN0.TMDF00 000A 8608h, RSCAN0.TMDF01 000A 8618h, RSCAN0.TMDF02 000A 8628h, RSCAN0.TMDF03 000A 8638h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDB1[7:0] | | | | | | | | TMDB0[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | TMDB0[7:0] | Transmit Buffer Data Byte 0 | Set transmit buffer data. | R/W |
| b15 to b8 | TMDB1[7:0] | Transmit Buffer Data Byte 1 | | R/W |

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

## 36.2.71 Transmit Buffer Register pCH (TMDF1p) (p = 0 to 3)

Address(es): RSCAN0.TMDF10 000A 860Ah, RSCAN0.TMDF11 000A 861Ah, RSCAN0.TMDF12 000A 862Ah, RSCAN0.TMDF13 000A 863Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDB3[7:0] | | | | | | | | TMDB2[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | TMDB2[7:0] | Transmit Buffer Data Byte 2 | Set transmit buffer data. | R/W |
| b15 to b8 | TMDB3[7:0] | Transmit Buffer Data Byte 3 | | R/W |

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

## 36.2.72 Transmit Buffer Register pDL (TMDF2p) (p = 0 to 3)

Address(es): RSCAN0.TMDF20 000A 860Ch, RSCAN0.TMDF21 000A 861Ch, RSCAN0.TMDF22 000A 862Ch, RSCAN0.TMDF23 000A 863Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDB5[7:0] | | | | | | | | TMDB4[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | TMDB4[7:0] | Transmit Buffer Data Byte 4 | Set transmit buffer data. | R/W |
| b15 to b8 | TMDB5[7:0] | Transmit Buffer Data Byte 5 | | R/W |

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

## 36.2.73 Transmit Buffer Register pDH (TMDF3p) (p = 0 to 3)

Address(es): RSCAN0.TMDF30 000A 860Eh, RSCAN0.TMDF31 000A 861Eh, RSCAN0.TMDF32 000A 862Eh, RSCAN0.TMDF33 000A 863Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDB7[7:0] | | | | | | | | TMDB6[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | TMDB6[7:0] | Transmit Buffer Data Byte 6 | Set transmit buffer data. | R/W |
| b15 to b8 | TMDB7[7:0] | Transmit Buffer Data Byte 7 | | R/W |

Modify this register when the corresponding TMSTSp.TMTRM flag is 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.
This register can be read/written when the GRWCR.RPAGE bit is 1.

## 36.2.74 Transmit History Buffer Control Register (THLCC0)

Address(es): RSCAN0.THLCC0 000A 837Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | THLDTE | THLIM | THLIE | — | — | — | — | — | — | — | THLE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | THLE | Transmit History Buffer Enable | 0: Transmit history buffer is not used.<br>1: Transmit history buffer is used. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | THLIE | Transmit History Interrupt Enable | 0: Transmit history interrupt is disabled.<br>1: Transmit history interrupt is enabled. | R/W |
| b9 | THLIM | Transmit History Interrupt Source Select | 0: When 6 sets of data have been stored in the transmit history buffer<br>1: When a single set of transmit history data has been stored | R/W |
| b10 | THLDTE | Transmit History Target Buffer Select | 0: Entry from transmit/receive FIFO buffers<br>1: Entry from transmit buffers, transmit/receive FIFO buffers | R/W |
| b15 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### THLE Bit (Transmit History Buffer Enable)

Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer. Modify this bit only in channel communication mode or channel halt mode.

### THLIE Bit (Transmit History Interrupt Enable)

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit with the THLE bit set to 0.

### THLIM Bit (Transmit History Interrupt Source Select)

This bit is used to select a transmit history interrupt source.
Modify this bit only in channel reset mode.

### THLDTE Bit (Transmit History Target Buffer Select)

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers and transmit/receive FIFO buffers is stored in the transmit history buffer.
Modify this bit only in channel reset mode.

## 36.2.75 Transmit History Buffer Status Register (THLSTS0)

Address(es): RSCAN0.THLSTS0 000A 8380h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | THLMC[3:0] | | | | — | — | — | — | THLIF | THLELT | THLFLL | THLEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | THLEMP | Transmit History Buffer Empty Status Flag | 0: Transmit history buffer contains unread data.<br>1: Transmit history buffer contains no unread data (buffer empty). | R |
| b1 | THLFLL | Transmit History Buffer Full Status Flag | 0: Transmit history buffer is not full.<br>1: Transmit history buffer is full. | R |
| b2 | THLELT | Transmit History Buffer Overflow Flag | 0: Transmit history buffer overflow has not occurred.<br>1: Transmit history buffer overflow has occurred. | R/(W) *1 |
| b3 | THLIF | Transmit History Interrupt Request Flag | 0: No transmit history interrupt request is present.<br>1: A transmit history interrupt request is present. | R/(W) *1 |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 to b8 | THLMC[3:0] | Transmit History Buffer Unread Data Counter | These bits indicate the number of unread data sets stored in the transmit history buffer. | R |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 0 can be written to this bit, to clear the flag. Writing 1 does not affect the flag value.

### THLEMP Flag (Transmit History Buffer Empty Status Flag)

The THLEMP flag becomes 0 when even a single set of transmit history data has been stored in the transmit history buffer.

This flag becomes 1 when all the data in the transmit history buffer has been read. This flag also becomes 1 in channel reset mode or when the THLCC0.THLE bit is set to 0 (transmit history buffer is not used).

### THLFLL Flag (Transmit History Buffer Full Status Flag)

The THLFLL flag becomes 1 when 8 data sets have been stored in the transmit history buffer, and becomes 0 when the number of data sets stored in the transmit history buffer has decreased to less than 8.

This flag also becomes 0 in channel reset mode or when the THLCC0.THLE bit is set to 0 (transmit history buffer is not used).

### THLELT Flag (Transmit History Buffer Overflow Flag)

The THLELT flag becomes 1 when it is attempted to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded.

This flag becomes 0 in channel reset mode or by writing 0 to this flag by the program.

### THLIF Flag (Transmit History Interrupt Request Flag)

The THLIF flag becomes 1 when the interrupt source set by the THLCC0.THLIM bit has occurred.

This flag becomes 0 in channel reset mode or by writing 0 to this flag by the program.

### THLMC[3:0] Flags (Transmit History Buffer Unread Data Counter)

These flags indicate the number of unread data sets stored in the transmit history buffer.

## 36.2.76 Transmit History Buffer Access Register (THLACC0)

Address(es): RSCAN0.THLACC0 000A 8680h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TID[7:0] | | | | | | | | — | — | — | BN[1:0] | | — | BT[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | BT[1:0] | Buffer Type Data | b1 b0<br>0 1: Transmit buffer<br>1 0: Transmit FIFO buffer | R |
| b2 | — | Reserved | This bit is read as 0. | R |
| b4, b3 | BN[1:0] | Buffer Number Data | The buffer number of transmit source (transmit buffer or transmit/receive FIFO) can be read. | R |
| b7 to b5 | — | Reserved | These bits are read as 0. | R |
| b15 to b8 | TID[7:0] | Label Data | The label information of stored data can be read. | R |

This register can be read when the GRWCR.RPAGE bit is 1.

### BT[1:0] Bits (Buffer Type Data)

These bits indicate the transmit source buffer type of transmit history data stored in the transmit history buffer.

### BN[1:0] Bits (Buffer Number Data)

These bits indicate the transmit source buffer number of transmit history data stored in the transmit history buffer.

### TID[7:0] Bits (Label Data)

These bits indicate the label information of transmit history data stored in the transmit history buffer.

## 36.2.77 Transmit History Buffer Pointer Control Register (THLPCTR0)

Address(es): RSCAN0.THLPCTR0 000A 8384h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | THLPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | THLPC[7:0] | Transmit History Buffer Pointer | Writing FFh to these bits moves the read pointer to the next unread data in the transmit history buffer. | W |
| b15 to b8 | — | Reserved | The write value should be 0. | W |

### THLPC[7:0] Bits (Transmit History Buffer Pointer)

When the THLPC [7:0] bits are set to FFh, the read pointer moves to the next data in the transmit history buffer. At this time, the THLSTS0.THLMC[3:0] (transmit history buffer unread data counter) value is decremented. After reading the THLACC0 register, write FFh to the THLPC [7:0] bits.

Write FFh to the THLPC[7:0] bits when the THLCC0.THLE bit is set to 1 (transmit history buffer is used) and the THLSTS0.THLEMP flag is 0.

## 36.2.78 Global RAM Window Control Register (GRWCR)

Address(es): RSCAN.GRWCR 000A 838Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RPAGE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RPAGE | RAM Window Select | 0: Selects window 0 (receive rule entry registers, RAM test registers)<br>1: Selects window 1 (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, transmit history data access register) | R/W |
| b15 to b1 | — | Reserved | The write value should be 0. | R/W |

### RPAGE Bit (RAM Window Select)

This bit is used to select a window for the switching of registers that are allocated to addresses from 000A 83A0h to 000A 8681h.

[Registers allocated when the RPAGE bit is set to 0 (window 0 selected)]

- Receive rule entry registers: GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, GAFLPHj (j = 0 to 15)
- RAM test registers: RPGACCr (r = 0 to 127)

[Registers allocated when the RPAGE bit is set to 1 (window 1 selected)]

- Receive buffer registers: RMIDLn, RMIDHn, RMTSn, RMPTRn, RMDF0n to RMDF3n (n = 0 to 15)
- Receive FIFO access registers: RFIDLm, RFIDHm, RFTSm, RFPTRm, RFDF0m to RFDF3m (m = 0, 1)
- Transmit/receive FIFO access registers: CFIDL0, CFIDH0, CFTS0, CFPTR0, CFDF00 to CFDF30
- Transmit buffer registers: TMIDLp, TMIDHp, TMPTRp, TMDF0p to TMDF3p (p = 0 to 3)
- Transmit history buffer access register: THLACC0

## 36.2.79 Global Test Configuration Register (GTSTCFG)

Address(es): RSCAN.GTSTCFG 000A 838Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | RTMPS[2:0] | | | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b10 to b8 | RTMPS[2:0] | RAM Test Page Configuration | Set a value within a range of page 0 (00h) to page 2 (02h). | R/W |
| b15 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Modify the GTSTCFG register only in global test mode.

### RTMPS[2:0] Bits (RAM Test Page Configuration)

These bits are used to set the RAM test target page number for RAM test. Set a value from 00h to 02h.

## 36.2.80 Global Test Control Register (GTSTCTRL)

Address(es): RSCAN.GTSTCTRL 000A 838Eh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | RTME | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b2 | RTME | RAM Test Enable | 0: RAM test is disabled.<br>1: RAM test is enabled. | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### RTME Bit (RAM Test Enable)

Setting this bit to 1 enables RAM test. Modify this bit only in global test mode.

(1) Set the GCTRL.GMDC[1:0] bits to 10b (global test mode).
(2) Unlock protection by successively writing 7575h and 8A8Ah to the GLOCKK register
(3) Set the RTME bit to 1.
(4) Check that the RTME bit is set to 1.

## 36.2.81 Global Test Protection Unlock Register (GLOCKK)

Address(es): RSCAN.GLOCKK 000A 8394h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LOCK[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | LOCK[15:0] | Protection Unlock Data | Write protection unlock data to use test functions. These bits are read as 0000h. | W |

Modify the GLOCKK register only in global test mode.

**LOCK[15:0] Bits (Protection Unlock Data)**

Write the protection unlock data shown in Table 36.3 to the LOCK[15:0] bits in succession to allow writing 1 to the target bit.

**Table 36.3 Protection Unlock Data for Test Functions**

| Test Function | Protection Unlock Data 1 | Protection Unlock Data 2 | Target Bit |
|---|---|---|---|
| RAM test | 7575h | 8A8Ah | GTSTCTRL.RTME bit |

Writing data to the CAN's register area (000A 8300h to 000A 839Fh) except the RAM area after protection is unlocked enables protection again.
Protection is not enabled even by reading data from the CAN's register area or reading/writing data from/to other areas.

## 36.2.82 RAM Test Register r (RPGACCr) (r = 0 to 127)

Address(es): RSCAN.RPGACC0 to RSCAN.RPGACC127 000A 8580h to 000A 867Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RDTA[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Description | R/W |
|---|---|
| Data can be read and written in CAN RAM. | R/W |

Modify the RPGACCr register in global test mode with the GTSTCTRL.RTME bit set to 1 (RAM test is enabled). The RPGACCr register is readable and writable when the GTSTCTRL.RTME bit is set to 1.
This register can be read/written when the GRWCR.RPAGE bit is 0.

## 36.3 CAN Modes

The CAN module has four global modes to control entire CAN module status and four channel modes to control individual channel status.

Details of global modes are described in section 36.3.1, Global Modes, and details of channel modes are described in section 36.3.2, Channel Modes.

- Global stop mode: Stops clocks of entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for entire module.
- Global test mode: Performs test settings and performs RAM test.
- Global operating mode: Makes entire module operable.
- Channel stop mode: Stops channel clock.
- Channel reset mode: Performs initial settings for channels.
- Channel halt mode: Stops CAN communication and enables channel test.
- Channel communication mode: Performs CAN communication.

### 36.3.1 Global Modes

Figure 36.2 shows the transitions of global modes.

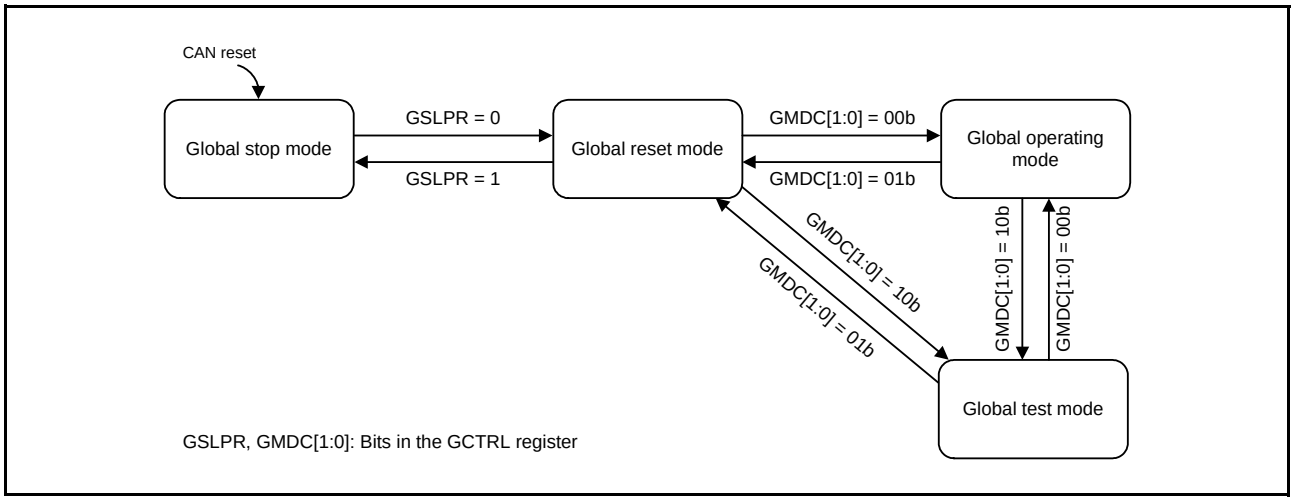



**Figure 36.2 Transitions of Global Modes**

Channel modes transition in some cases with transitions of global modes. Table 36.4 shows the transitions of channel modes depending on the global mode setting by the GCTRL.GMDC[1:0] bits and the GSLPR bit.

**Table 36.4 Transitions of Channel Modes Depending on Global Mode Setting (GCTRL.GMDC[1:0] and GSLPR Bits)**

| | Channel Mode after Setting | | | |
|---|---|---|---|---|
| **Channel Mode before Setting** | **GMDC[1:0] = 00b GSLPR = 0 (Global Operation)** | **GMDC[1:0] = 10b GSLPR = 0 (Global Test)** | **GMDC[1:0] = 01b GSLPR = 0 (Global Reset)** | **GMDC[1:0] = 01b GSLPR = 1 (Global Stop)** |
| Channel communication | Channel communication | Channel communication | Channel reset | Transition prohibited |
| Channel halt | Channel halt | Channel halt | Channel reset | Transition prohibited |
| Channel reset | Channel reset | Channel reset | Channel reset | Channel stop |
| Channel stop | Channel stop | Channel stop | Channel stop | Channel stop |

Table 36.5 shows the global mode transition time.

**Table 36.5 Global Mode Transition Time**

| Mode before Transition | Mode after Transition | Maximum Transition Time |
|---|---|---|
| Global stop | Global reset | 3 PCLK cycles |
| Global reset | Global stop | 3 PCLK cycles |
| Global reset | Global test | 10 PCLK cycles |
| Global reset | Global operating | 10 PCLK cycles |
| Global test | Global reset | 3 PCLK cycles |
| Global test | Global operating | 3 PCLK cycles |
| Global operating | Global reset | 3 PCLK cycles |
| Global operating | Global test | Two CAN frames |

### (1) Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.
After the operation of the CAN module is enabled, the CAN module transitions to global stop mode. Setting the GCTRL.GSLPR bit to 1 (global stop mode) in global reset mode sets the CTRL.CSLPR bit to 1 (channel stop mode). If all channels are forcibly caused to transition to channel stop mode, the CAN module transitions to global stop mode. The GCTRL.GSLPR bit should not be modified in global operating mode and global test mode.

### (2) Global Reset Mode

In global reset mode, CAN module settings are performed. When the CAN module transitions to global reset mode, some registers are initialized. Table 36.8 and Table 36.9 list the registers to be initialized.
Setting the GCTRL.GMDC[1:0] bits to 01b sets each of the CTRL.CHMDC[1:0] bits to 01b (channel reset mode). If all channels are forcibly caused to transition to channel reset mode, the CAN module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CTRL.CHMDC[1:0] bits have already been set to 01b).

### (3) Global Test Mode

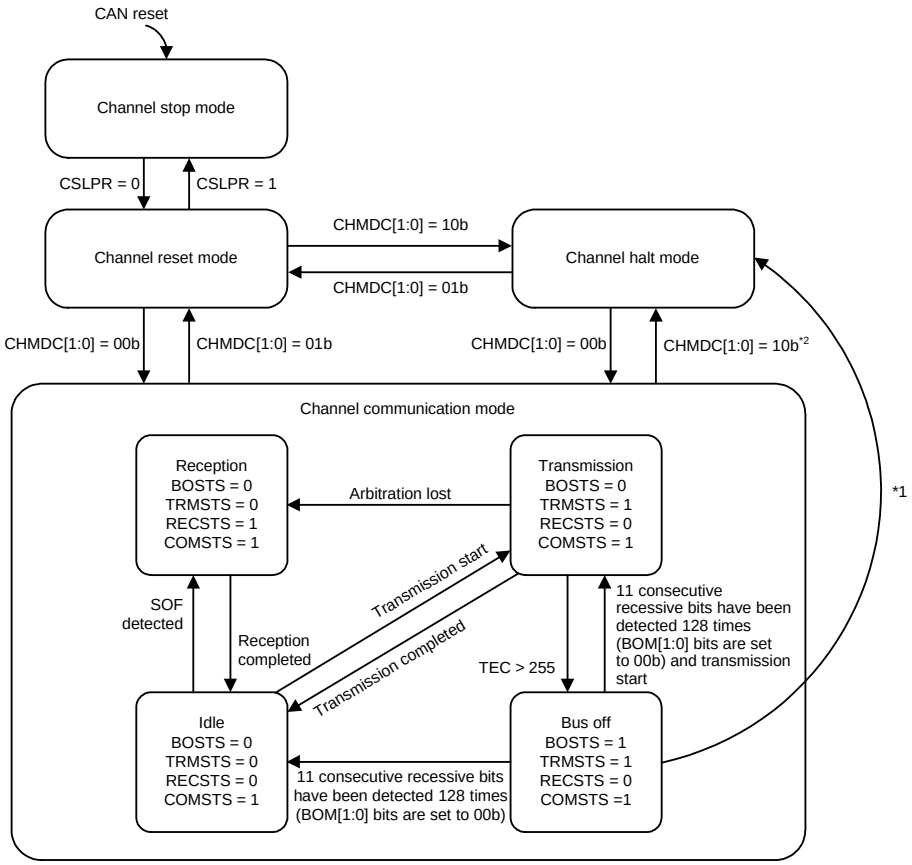
In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.
Setting the GCTRL.GMDC[1:0] bits to 10b sets each of the CTRL.CHMDC[1:0] bits to 10b (channel halt mode). If all channels are forcibly caused to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

### (4) Global Operating Mode

In global operating mode, entire CAN module operates.
When the GCTRL.GMDC[1:0] bits are set to 00b, the CAN module transitions to global operating mode.

## 36.3.2 Channel Modes

Figure 36.3 shows a channel mode state transition chart. Table 36.6 shows the channel mode transition time.




CHMDC[1:0], CSLPR: Bits in the CTRL register
BOM[1:0]: Bits in the CTRH register
BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the STSL register

Note 1. Timing of transition from bus off state to channel halt mode
- When BOM[1:0] = 01b: Transition to channel halt mode when TEC exceeds 255
- When BOM[1:0] = 10b: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
- When BOM[1:0] = 11b: Transition to channel halt mode when the CHMDC[1:0] bits are set to 10b

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode.

**Figure 36.3 Channel Mode State Transition Chart**

**Table 36.6 Channel Mode Transition Time**

| Mode before Transition | Mode after Transition | Maximum Transition Time |
|---|---|---|
| Channel stop | Channel reset | 3 PCLK cycles |
| Channel reset | Channel stop | 3 PCLK cycles |
| Channel reset | Channel halt | 3 CAN bit times |
| Channel reset | Channel communication | 2 CAN bit times |
| Channel halt | Channel reset | 3 PCLK cycles |
| Channel halt | Channel communication | 3 CAN bit times |
| Channel communication | Channel reset | 3 PCLK cycles |
| Channel communication | Channel halt | 2 CAN frames |

### (1) Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.
Each channel enters channel stop mode after the operation of the CAN module is enabled. The channel transitions to channel stop mode when the CTRL.CSLPR bit is set to 1 (channel stop mode) in channel reset mode.
The CSLPR bit should not be modified in channel communication mode and channel halt mode.

### (2) Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. Table 36.8 lists the registers to be initialized.
When the CTRL.CHMDC[1:0] bits are set to 01b (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. Table 36.7 shows the operation when the CTRL.CHMDC[1:0] bits are set to 01b (channel reset mode) during CAN communication.

### (3) Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.
Table 36.7 shows operation when the CTRL.CHMDC[1:0] bits are set to 10b (channel halt mode) during CAN communication.

**Table 36.7 Operation when a Channel Transitions to Channel Reset Mode/Channel Halt Mode**

| Mode | During Reception | During Transmission | Bus Off State |
|---|---|---|---|
| Channel reset (CHMDC[1:0] = 01b) | Transitions to channel reset mode before reception is completed.*1 | Transitions to channel reset mode before transmission is completed.*1 | Transitions to channel reset mode before bus off recovery. |
| Channel halt*3 (CHMDC[1:0] = 10b) | Transitions to channel halt mode after reception is completed.*2 | Transitions to channel halt mode after transmission is completed.*2 | [When BOM[1:0] = 00b]<br>Transitions to channel halt mode (CHMDC[1:0] = 10b) only after bus off recovery.<br>[When BOM[1:0] = 01b]<br>Transitions to channel halt mode automatically when the condition for transition to bus off state is met.<br>[When BOM[1:0] = 10b]<br>Transitions to channel halt mode automatically after bus off recovery.<br>[When BOM[1:0] = 11b]<br>Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10b before bus off recovery. |

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to 10b and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01b.
Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the ERFLL.BLF flag that becomes 1 when dominant lock is detected.
Note 3. In case of a transition from channel reset mode to channel halt mode, transition to channel halt mode after setting the CFGL and CFGH registers in channel reset mode.

### (4) Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle: Neither reception nor transmission is in progress.
- Reception: Receiving a message sent from another node.
- Transmission: Transmitting a message.

- Bus off: Isolated from CAN communication.

When the CTRL.CHMDC[1:0] bits are set to 00b, the channel transitions to channel communication mode. After that, when 11 consecutive recessive bits have been detected, the STSL.COMSTS flag becomes 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

### (5) Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the ISO 11898-1 standard.
How to return from the bus off state is set by the CTRH.BOM[1:0] bits.

- When CTRH.BOM[1:0] = 00b:
  Bus off recovery is compliant with the ISO 11898-1 standard. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state).
  At that time, the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h and the ERFLL.BORF flag becomes 1 (bus off recovery is detected). When the CTRL.CHMDC[1:0] bits are set to 10b (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When CTRH.BOM[1:0] = 01b:
  When a channel transitions to the bus off state, the CTRL.CHMDC[1:0] bits are set to 10b and the channel transitions to channel halt mode. At that time, the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h but the ERFLL.BORF flag is not set to 1.
- When CTRH.BOM[1:0] = 10b:
  When a channel has transitioned to the bus off state, the CTRL.CHMDC[1:0] bits are set to 10b. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h and the ERFLL.BORF flag becomes 1.
- When CTRH.BOM[1:0] = 11b:
  When the CHMDC[1:0] bits are set to 10b in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h but the ERFLL.BORF flag is not set to 1.
  However, the BORF flag becomes 1 if a CAN module transitions to error active state (by detecting 128 times of 11 consecutive recessive bits) before the CTRL.CHMDC[1:0] bits are set to 10b.

If the channel transitions to channel halt mode simultaneously when the program writes a value to the CTRL.CHMDC[1:0] bits, writing by the program takes precedence. An automatic transition to channel halt mode when the CTRH.BOM[1:0] bits are set to 01b or 10b is made only when the CTRL.CHMDC[1:0] bits are 00b (channel communication mode).
Furthermore, setting the CTRL.RTBO bit to 1 allows forcible return from the bus off state. As soon as the CTRL.RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the condition of CAN module becomes ready for communication. In this case, the ERFLL.BORF flag is not set to 1 and the STSH.TEC[7:0] and STSH.REC[7:0] flags are initialized to 00h. Write 1 to the CTRL.RTBO bit when the CTRH.BOM[1:0] value is 00b.

**Table 36.8 Registers Initialized in Global Reset Mode or Channel Reset Mode**

| Register | Bit/Flag |
|---|---|
| CTRL | CHMDC[1:0] |
| CTRH | CTMS[1:0], CTME |
| STSL | CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS |
| STSH | REC[7:0], TEC[7:0] |
| ERFLL | ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF |
| ERFLH | CRCREG[14:0] |
| CFCCL0 | When transmit/receive FIFO buffer is in transmit mode: CFE |
| CFSTS0 | When transmit/receive FIFO buffer is in transmit mode: CFMC[5:0], CFTXIF, CFRXIF, CFMLT, CFFLL, CFEMP |
| TMCp | TMOM, TMTAR, TMTR |
| TMSTSp | TMTARM, TMTRM, TMTRF[1:0], TMTSTS |
| TMTRSTS | TMTRSTSp |
| TMTCSTS | TMTCSTSp |
| TMTASTS | TMTASTSp |
| THLCC0 | THLE |
| THLSTS0 | THLMC[3:0], THLIF, THLELT, THLFLL, THLEMP |
| GTINTSTS | THIF0, CFTIF0, TAIF0, TSIF0 |

**Table 36.9 Registers Initialized Only in Global Reset Mode**

| Register | Bit/Flag |
|---|---|
| GSTS | GHLTSTS |
| GERFLL | THLES, MES, DEF |
| GTSC | TS[15:0] |
| RMND0 | RMNSn |
| RFCCm | RFE |
| RFSTSm | RFMC[5:0], RFIF, RFMLT, RFFLL, RFEMP |
| CFCCL0 | When transmit/receive FIFO buffer is in receive mode: CFE |
| CFSTS0 | When transmit/receive FIFO buffer is in receive mode: CFMC[5:0], CFTXIF, CFRXIF, CFMLT, CFFLL, CFEMP |
| RFMSTS | RFmMLT |
| CFMSTS | CF0MLT |
| RFISTS | RFmIF |
| CFISTS | CF0IF |
| GTSTCFG | RTMPS[2:0] |
| GTSTCTRL | RTME |

## 36.4 Reception Function

There are two reception types.

- Reception by receive buffers:
  Zero to 16 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):
  Two receive FIFO buffers can be shared by all channels and one dedicated transmit/receive FIFO buffer is provided for each channel. The FIFO buffers can hold the number of received messages set by the RFCCm.RFDC[2:0] bits and CFCCL0.CFDC[2:0] bits, and messages can be read sequentially from the oldest.

### 36.4.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows selected messages to be stored in the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.
Up to 16 receive rules can be registered per channel. If receive rules are not set, no message can be received. Figure 36.4 illustrates how receive rules are registered.

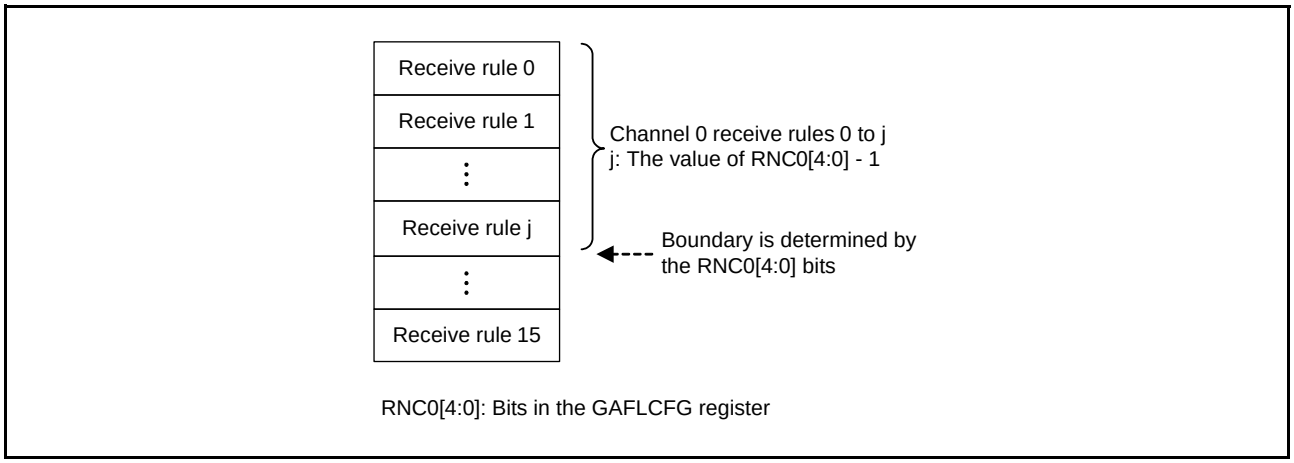



**Figure 36.4 Entry of Receive Rules**

Each receive rule consists of 12 bytes in the GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, and GAFLPHj registers. The GAFLIDLj and GAFLIDHj registers are used to set ID, IDE bit, RTR bit, and the mirror function, the GAFLMLj and GAFLMHj registers are used to set mask, the GAFLPLj and GAFLPHj registers are used to set label information to be added, DLC value, and storage receive buffer, and storage FIFO buffer.

#### (1) Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in a received message which correspond to bits that are set to 0 (bits are not compared) in the GAFLMLj and GAFLMHj registers are not compared and are regarded as matched.
Check begins with the receive rule with the smallest rule number of the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

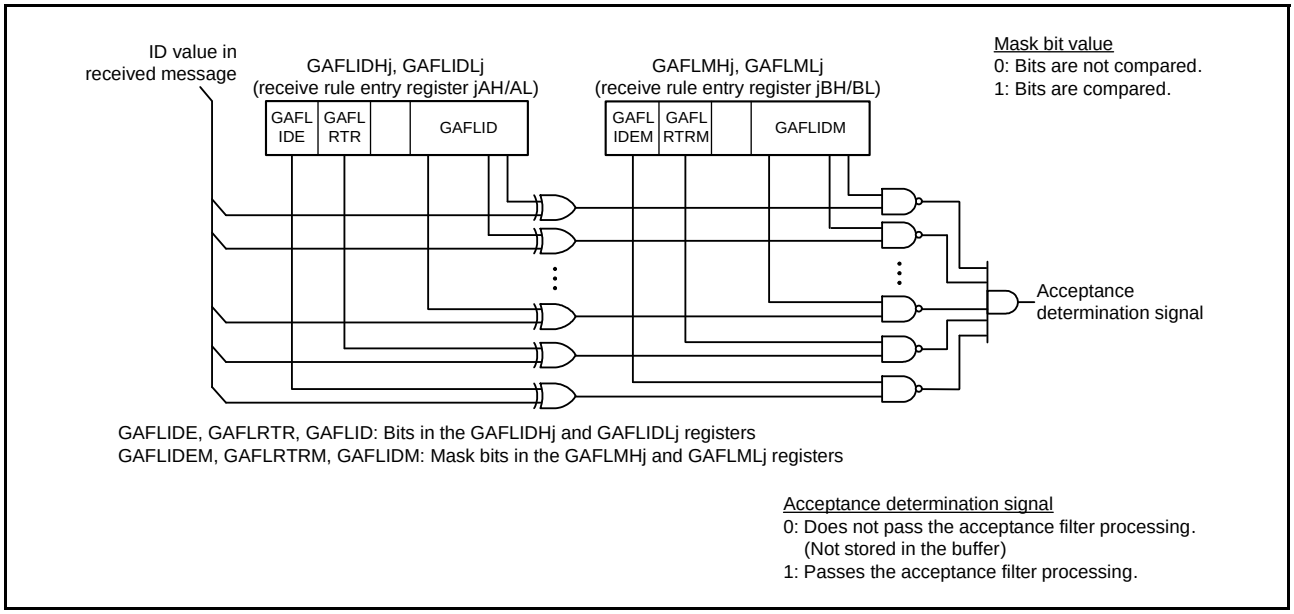



**Figure 36.5 Acceptance Filter Function**

### (2) DLC Filter Processing

When the GCFGL.DCE bit is set to 1 (DLC check is enabled), DLC filter processing is added to messages that passed through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the GCFGL.DRE bit set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the GCFGL.DRE bit set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of 00h is written to data bytes that are larger than the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the GERFLL.DEF flag becomes 1 (a DLC error is present).

### (3) Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers (set to receive mode). Message storage destination is set by the GAFLPLj.GAFLRMV, GAFLRMDP[6:0], GAFLFDP4, GAFLFDP1, and GAFLFDP0 bits. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to two buffers.

### (4) Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPHj.GAFLPTR[11:0] bits.

### (5) Mirror Function Processing

The mirror function allows reception of messages transmitted from the own CAN node. The mirror function is made available by setting the GCFGL.MME bit to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLIDHj.GAFLLB bit is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When receiving messages transmitted from the own CAN node, receive rules for which the GAFLIDHj.GAFLLB bit is set to 1 are used for data processing.

### 36.4.2 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time. The timestamp counter value is fetched at the start-of-frame (SOF) timing of a message and is then stored in a receive buffer or a FIFO buffer together with the message ID and data. PCLK or the CAN bit time clock is selectable as a timestamp counter clock source from the GCFGL.TSSS bit. The clock obtained by dividing the selected clock source by the GCFGL.TSP[3:0] value is used as the timestamp counter count source.

When the CAN bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When PCLK is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000h by setting the GCTRH.TSRST bit to 1.

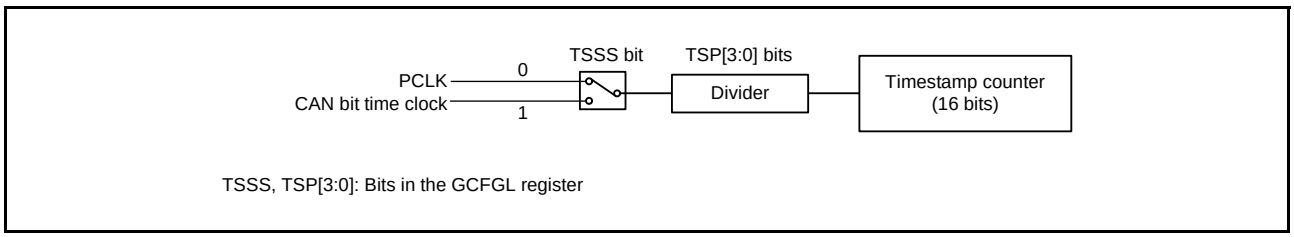



**Figure 36.6 Timestamp Function Block Diagram**

## 36.5 Transmission Functions

There are two types of transmission.

- Transmission using transmit buffers:
  Each channel has 4 buffers.
- Transmission using transmit/receive FIFO buffers (transmit mode):
  Each channel has one FIFO buffer. Up to 16 messages can be contained in a single FIFO buffer. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.

Figure 36.7 shows the allocation of transmit/receive FIFO buffer link.

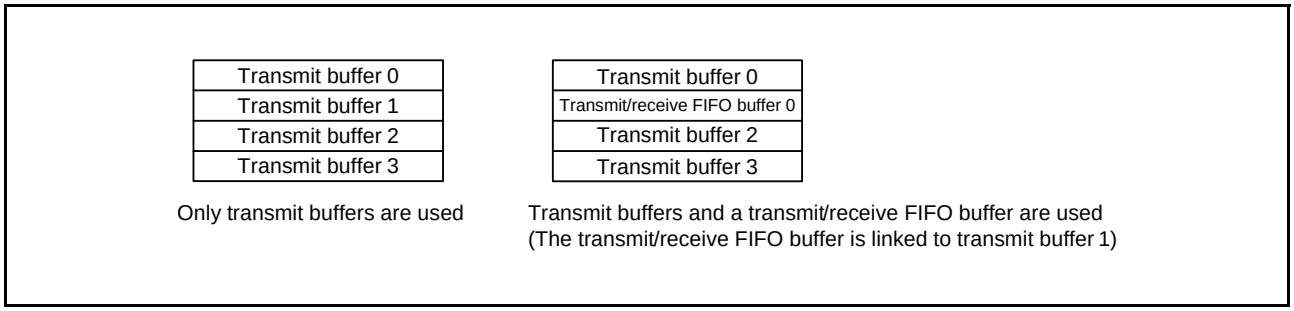



**Figure 36.7 Allocation of Transmit/Receive FIFO Buffer Links**

### 36.5.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers in the same channel, transmit priority is determined.
The priority is determined by using one of the following methods.

- ID priority (GCFGL.TPRI bit = 0)
- Transmit buffer number priority (GCFGL.TPRI bit = 1)

The setting of the GCFGL.TPRI bit is enabled in all CAN channels.
When the GCFGL.TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. Priority of IDs conforms to the CAN bus arbitration specification defined in the ISO 11898-1 standard. IDs of messages stored in transmit buffers and transmit/receive FIFO buffers (set to transmit mode) are targets of priority determination. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When the GCFGL.TPRI bit is set to 1, the message in the transmit buffer of the minimum number among buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.
When messages are retransmitted due to an arbitration lost or an error, transmit priority determination is made again regardless of the GCFGL.TPRI bit setting.

### 36.5.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMCp.TMTR bit) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.
Transmit result is shown by the corresponding TMSTSp.TMTRF[1:0] flags. When transmit completes successfully, the TMSTSp.TMTRF[1:0] flags become 10b (transmission has been completed (without transmit abort request)) or 11b (transmission has been completed (with transmit abort request)).

#### (1) Transmit Abort Function

With respect to transmit buffers for which the TMSTSp.TMTRM flag is 1 (a transmit request is present), when the TMCp.TMTAR bit is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMSTSp.TMTRF[1:0] flags become 01b (transmit abort has been completed) and the transmit request is canceled (the TMSTSp.TMTRM flag becomes 0).
A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration lost or an error has occurred while a message for which the TMCp.TMTAR bit is set to 1 is being transmitted, retransmission is not performed.

#### (2) One-Shot Transmission Function (Retransmission Disabling Function)

When the TMCp.TMOM bit is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration lost or an error occurs, retransmission is not performed.
One-shot transmit result is shown by the corresponding TMSTSp.TMTRF[1:0] flags. When one-shot transmit completes successfully, the TMSTSp.TMTRF[1:0] flags become 10b or 11b. When an arbitration lost or an error has occurred, the TMSTSp.TMTRF[1:0] flags become 01b (transmit abort has been completed).

### 36.5.3 Transmission Using FIFO Buffers

Messages of a volume of the FIFO buffer depth set by the CFCCL0.CFDC[2:0] bits can be stored in a single transmit/receive FIFO buffer. Messages are transmitted sequentially on a first-in, first-out basis.
Transmit/receive FIFO buffers are linked to transmit buffers selected by the CFCCH0.CFTML[1:0] bits.
When the CFCCL0.CFE bit is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority determination is made for only the message to be transmitted next in a FIFO buffer.
When the CFCCL0.CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFSTS0.CFEMP flag becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration lost when the message in it is being transmitted or to be transmitted next.

When the CFCCL0.CFE bit is set to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFSTS0.CFEMP flag becomes 1 before setting the CFCCL0.CFE bit to 1 again.

#### (1) Interval Transmission Function

To transmit messages from the same FIFO buffer while a transmit/receive FIFO buffer that is set to transmit mode is in use, message transmission interval time can be set.
Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFCCL0.CFE bit set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by setting the CFCCL0.CFE bit to 0.
The interval time is set by the CFCCH0.CFITT[7:0] bits. When the interval timer is not used, set the

CFCCH0.CFITT[7:0] bits to 00h.
Select an interval timer count source by the CFCCH0.CFITR and CFITSS bits. When the CFCCH0.CFITR and CFITSS bits are set to 00b, the clock obtained by frequency-dividing PCLK by the GCFGH.ITRCP[15:0] value is used as a count source. When the CFCCH0.CFITR and CFITSS bits are set to 10b, the clock obtained by frequency-dividing PCLK by the GCFGH.ITRCP[15:0] value × 10 is used as a count source. When the CFCCH0.CFITR and CFITSS bits are set to x1b, the CAN bit time clock is used as a count source.
The interval time is calculated by the following equations where M is the set GCFGH.ITRCP[15:0] value and N is the set CFCCH0.CFITT[7:0] value.

- When CFCCH0.CFITR and CFITSS = 00b

$$\frac{1}{PCLK} \times M \times N$$

- When CFCCH0.CFITR and CFITSS = 10b

$$\frac{1}{PCLK} \times M \times 10 \times N$$

- When CFCCH0.CFITR and CFITSS = x1b
(fCANBIT is CAN bit time clock frequency)

$$\frac{1}{fCANBIT} \times N$$

Figure 36.8 shows the interval timer block diagram.

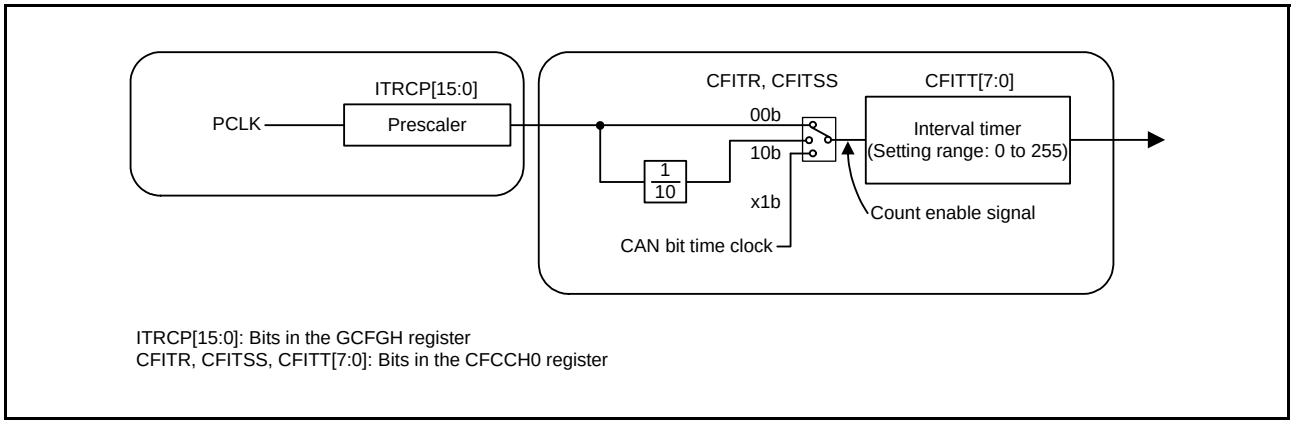



**Figure 36.8 Interval Timer Block Diagram**

Figure 36.9 shows the interval timer timing chart.

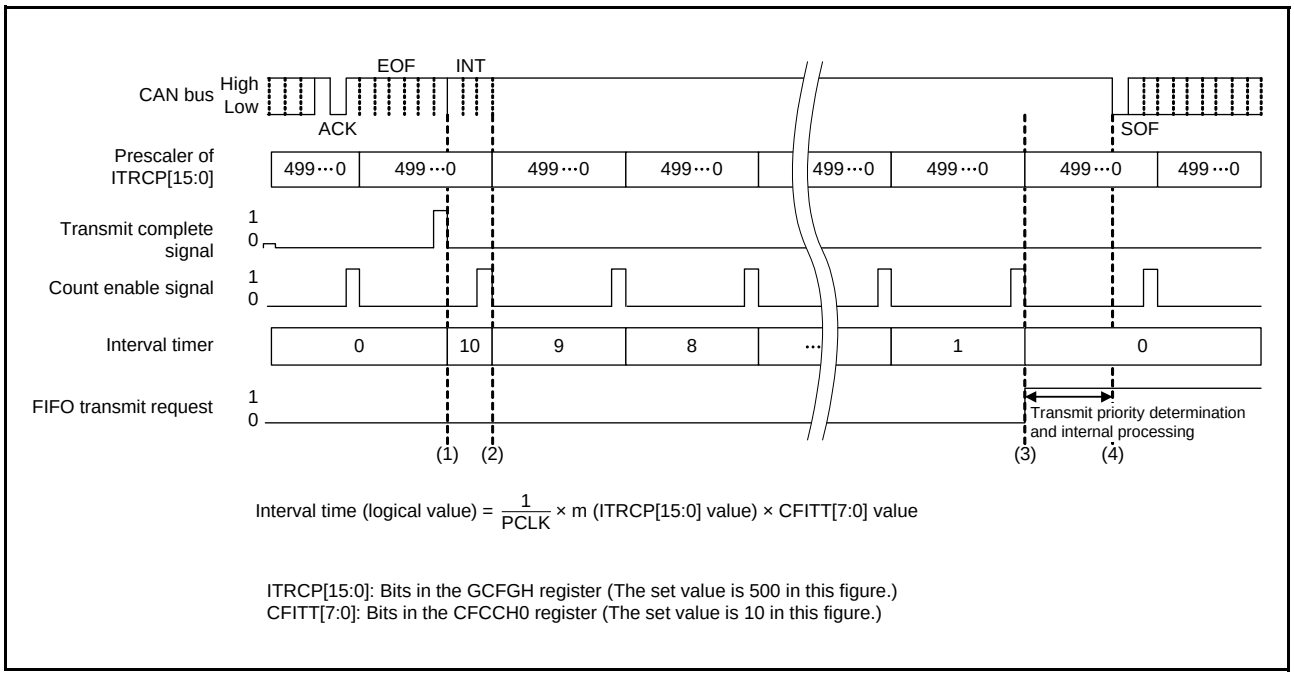



**Figure 36.9 Interval Timer Timing Chart**

(1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.
(2) The interval timer is decremented by the next count enable signal.
(3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.
(4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts with a delay of three CAN bit time clock cycles or less from the issue of transmit request.

### 36.5.4 Transmit History Function

Information of transmitted messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 8 sets of transmit history data.
A message transmit source buffer type can be selected by the THLCC0.THLDTE bit. Whether to store transmit history data for each message can be set by the CFIDH0.THLEN bit.
After transmit completes successfully, information of the following transmit messages is stored in the transmit history buffer as transmit history data. After successful completion of transmit, process may be delayed by up to 38 clocks of PCLK before the transmit history data is stored.

- Buffer type 01b: Transmit buffer
  10b: Transmit/receive FIFO buffer
- Buffer number Number of source transmit buffer or transmit/receive FIFO buffer.
  This number depends on buffer types. See Table 36.10.
- Label data Label information of transmit message

**Table 36.10 Transmit History Data Buffer Numbers**

| Buffer Number | Buffer Type | |
|---|---|---|
| | 01b | 10b |
| 00b | Transmit buffer 0 | Numbers of transmit buffers linked to transmit/receive FIFO buffers by the CFCCH0.CFTML[1:0] bits. |
| 01b | Transmit buffer 1 | |
| 10b | Transmit buffer 2 | |
| 11b | Transmit buffer 3 | |

Label data is used to identify each message. A unique label data can be added to each message transmitted from a transmit buffer or transmit/receive FIFO buffer.
Transmit history data can be read from the THLACC0 register. If it is attempted to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

## 36.6 Test Function

The test function is classified into communication tests and global tests.
Communication tests: Performed for each channel.

- Standard test mode
- Listen-only mode
- Self-test mode 0 (external loopback mode)
- Self-test mode 1 (internal loopback mode)
  Global tests: Performed in entire module
- RAM test (read/write test)

### 36.6.1 Standard Test Mode

Standard test mode allows CRC test.

### 36.6.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted. Listen-only mode is available for detecting the communication speed.
Do not make a transmit request from any buffer in listen-only mode.
Figure 36.10 shows the connection when listen-only mode is selected.

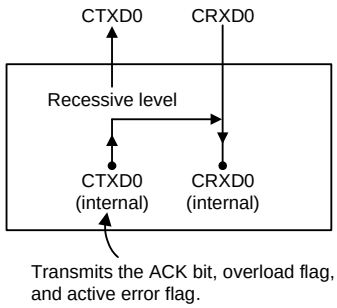



**Figure 36.10 Connection When Listen-Only Mode is Selected**

### 36.6.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLIDHj.GAFLLB bit is set to 0 (when a message transmitted from another CAN node is received).
If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

#### (1) Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.
In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.
Figure 36.11 shows the connection when self-test mode 0 is selected.

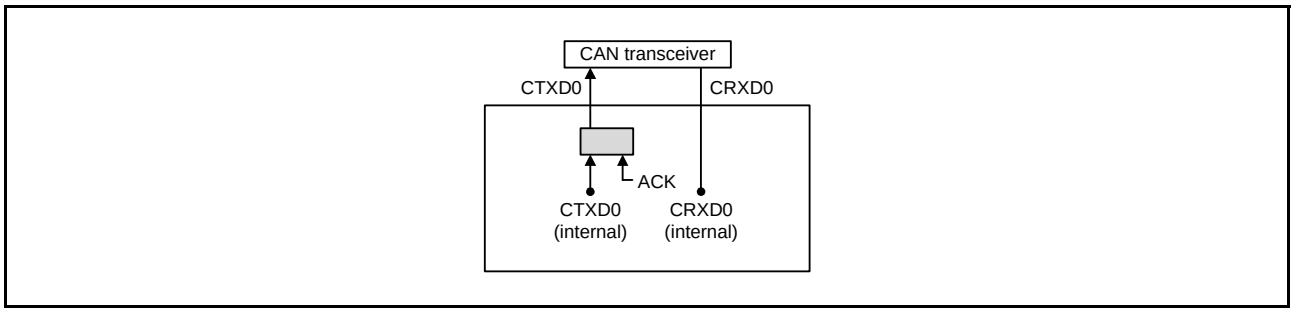



**Figure 36.11 Connection When Self-Test Mode 0 is Selected**

#### (2) Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.
In self-test mode 1, internal feedback from the internal CTXD0 pin to the internal CRXD0 pin is performed. The external CRXD0 pin input is isolated. The external CTXD0 pin outputs only recessive bits.
Figure 36.12 shows the connection when self-test mode 1 is selected.

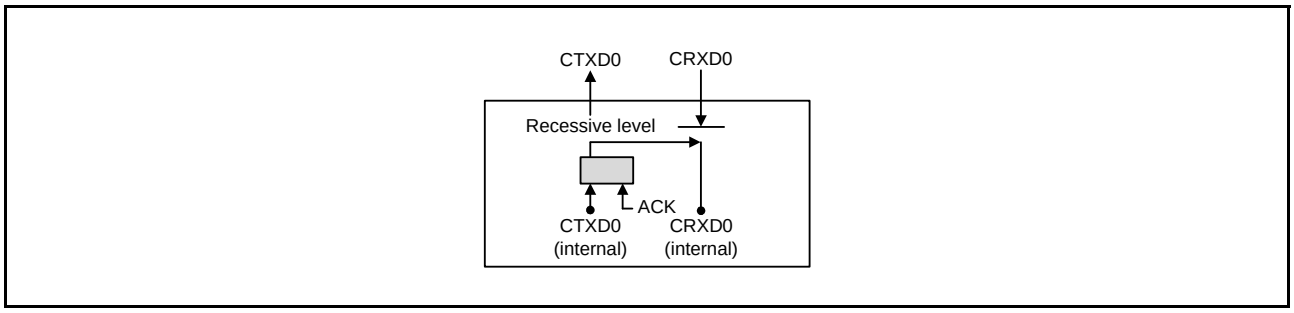



**Figure 36.12 Connection When Self-Test Mode 1 is Selected**

### 36.6.4 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.
When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test page is set by the GTSTCFG.RTMPS[2:0] bits. Data in the set page can be read from and written to the RPGACCr register. The available total RAM size is 544 bytes (0220h).

## 36.7 Interrupt

The CAN module has 5 interrupts that are grouped into global interrupts and channel interrupts.

Global interrupts (2 interrupts):

- Global receive FIFO interrupt (RXFINT)
- Global error interrupt (GLERRINT)

Channel interrupts (3 interrupts per channel):

- Channel transmit interrupt (TXINT)
  - Transmit complete interrupt
  - Transmit abort interrupt
  - Transmit/receive FIFO transmit complete interrupt (transmit mode)
  - Transmit history interrupt
- Transmit/receive FIFO receive interrupt (COMFRXINT)
- Channel error interrupt (CHERRINT)

When an interrupt request is generated, the corresponding CAN module interrupt request flag becomes 1 (interrupt request present). In that case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the CAN module. (Generation of interrupts also is controlled by the interrupt function.)
Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The next interrupt request is not generated until the interrupt request is cleared.

For details on the setting of the interrupt functions, refer to section 15, Interrupt Controller (ICUb).
In the following pages, Table 36.11 lists the CAN interrupt sources, Figure 36.13 shows the CAN global interrupt block diagram, and Figure 36.14 shows the CAN channel interrupt block diagram.

**Table 36.11 List of CAN Interrupt Sources**

| Item | Interrupt Source | | Corresponding Interrupt Request Flag*1 | Corresponding Interrupt Enable Bit *1 |
|---|---|---|---|---|
| Global interrupts | Global receive FIFO | Receive FIFO 0 | RFSTS0.RFIF | RFCC0.RFIE |
| | | Receive FIFO 1 | RFSTS1.RFIF | RFCC1.RFIE |
| | Global error | | GERFLL.DEF | GCTRL.DEIE |
| | | | GERFLL.MES | GCTRL.MEIE |
| | | | GERFLL.THLES | GCTRL.THLEIE |
| Channel interrupts | Channel transmit | Transmit complete | TMSTSp.TMTRF[1:0] | TMIEC.TMIEp |
| | | Transmit abort | TMSTSp.TMTRF[1:0] | CTRH.TAIE |
| | | Transmit/receive FIFO transmit | CFSTS0.CFTXIF | CFCCL0.CFTXIE |
| | | Transmit history | THLSTS0.THLIF | THLCC0.THLIE |
| | Transmit/receive FIFO receive | | CFSTS0.CFRXIF | CFCCL0.CFRXIE |
| | Channel error | | ERFLL.BEF | CTRL.BEIE |
| | | | ERFLL.ALF | CTRL.ALIE |
| | | | ERFLL.BLF | CTRL.BLIE |
| | | | ERFLL.OVLF | CTRL.OLIE |
| | | | ERFLL.BORF | CTRL.BORIE |
| | | | ERFLL.BOEF | CTRL.BOEIE |
| | | | ERFLL.EPF | CTRL.EPIE |
| | | | ERFLL.EWF | CTRL.EWIE |
| | Wakeup | | None | None |

Note 1. For details on the interrupt request flags and interrupt enable bits, refer to section 15, Interrupt Controller (ICUb).

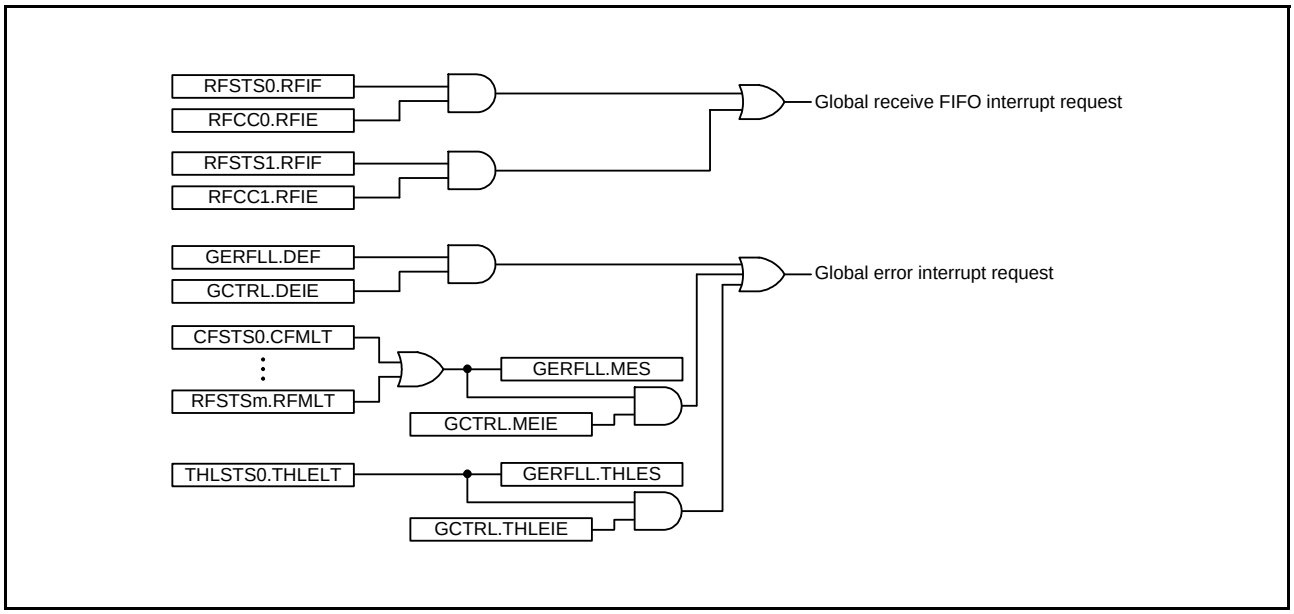



**Figure 36.13 CAN Global Interrupt Block Diagram**

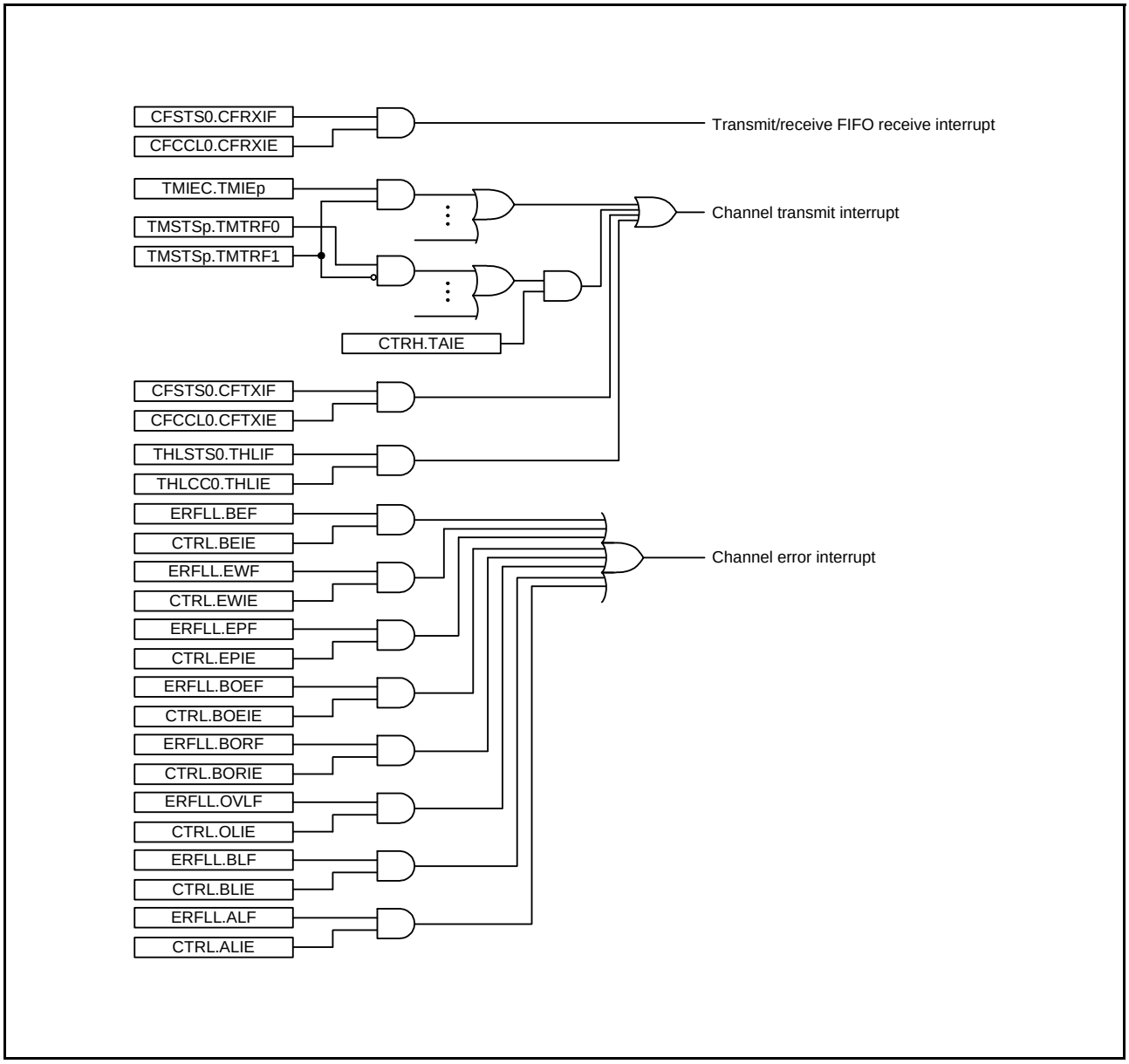



**Figure 36.14 CAN Channel Interrupt Block Diagram**

## 36.8 RAM Window

The CAN area from 000A 83A0h to 000A 8681h consists of two windows. The GRWCR.RPAGE bit is used to switch the allocation of registers.

- Registers allocated when the GRWCR.RPAGE bit is set to 0 (window 0 selected)
  Receive rule entry registers: GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, GAFLPHj
  RAM test registers: RPGACCr

- Registers allocated when the GRWCR.RPAGE bit is set to 1 (window 1 selected)
  Receive buffer registers: RMIDLn, RMIDHn, RMTSn, RMPTRn, RMDF0n to RMDF3n
  Receive FIFO access registers: RFIDLm, RFIDHm, RFTSm, RFPTRm, RFDF0m to RFDF3m
  Transmit/receive FIFO access registers: CFIDL0, CFIDH0, CFTS0, CFPTR0, CFDF00 to CFDF30
  Transmit buffer registers: TMIDLp, TMIDHp, TMPTRp, TMDF0p to TMDF3p
  Transmit history buffer access register: THLACC0

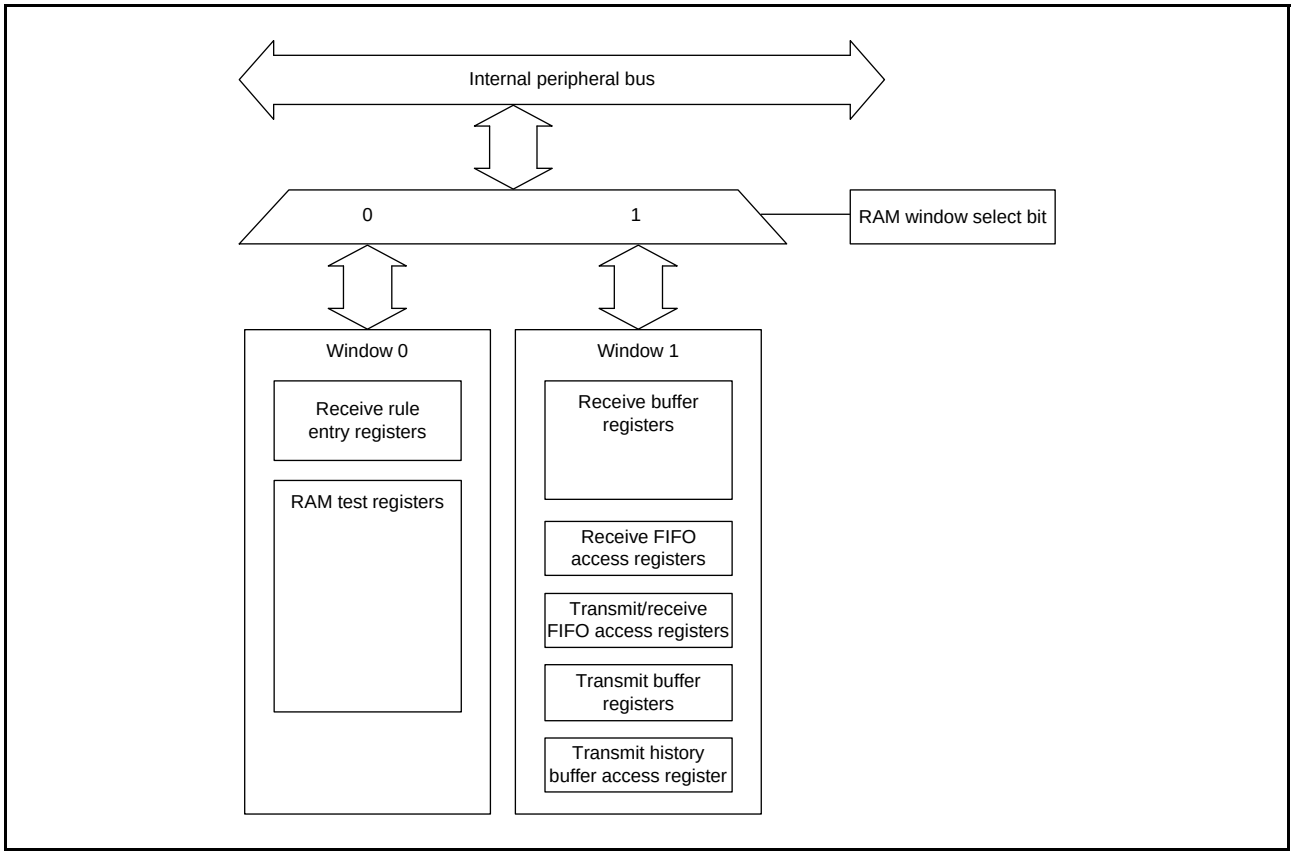
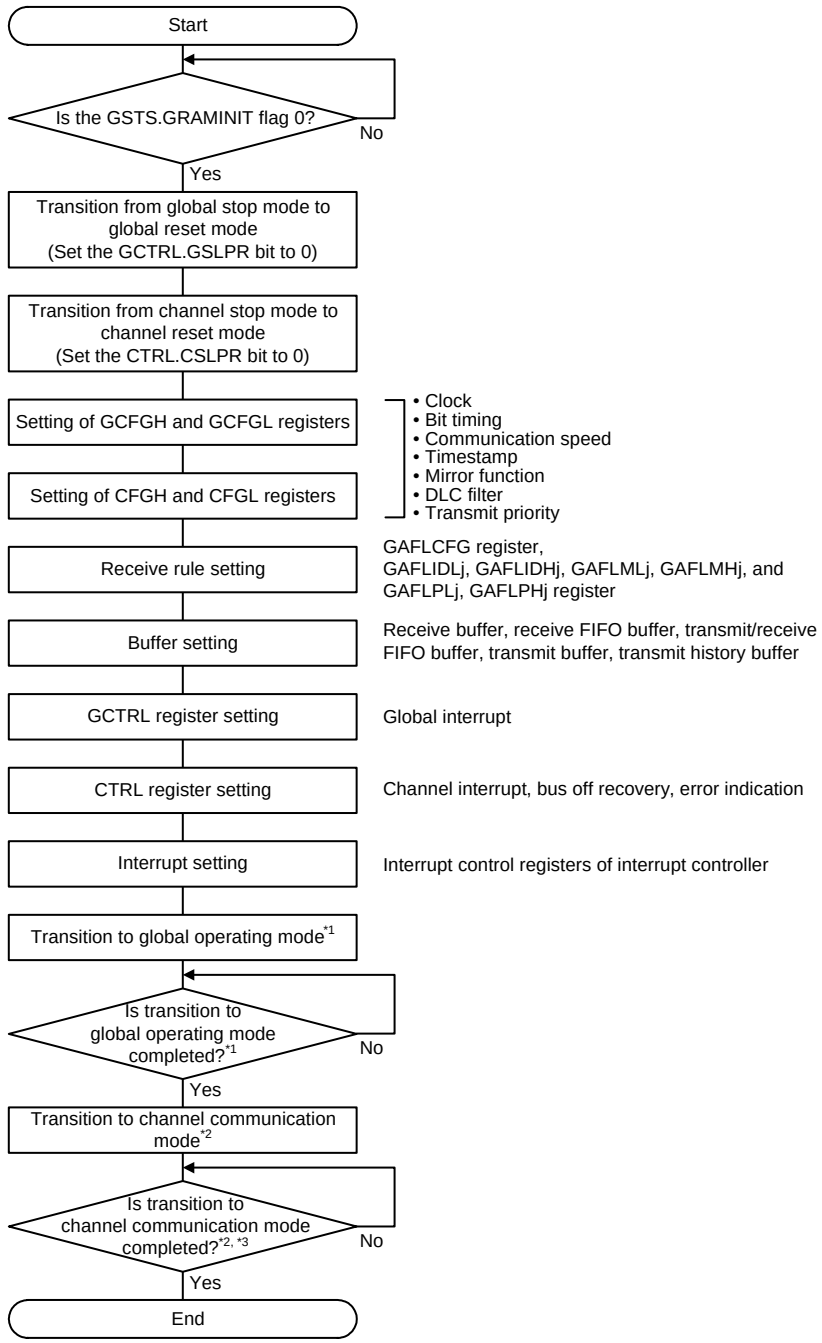



**Figure 36.15 RAM Window**

## 36.9 Initial Settings

The CAN module initializes the CAN RAM after the operation of the CAN module is enabled. The RAM initialization time is 276 cycles of PCLK. The GSTS.GRAMINIT flag becomes 1 (CAN RAM initialization is ongoing) during the RAM initialization and becomes 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GSTS.GRAMINIT flag becomes 0.

Figure 36.16 shows the CAN setting procedure after the operation of the CAN module is enabled.




Note 1. When global mode is changed by the GCTRL.GSLPR and GMDC[1:0] bits, use the GSTS register to confirm that the mode is switched. Do not modify the GCTRL.GMDC[1:0] bits until the mode is switched.

Note 2. When channel mode is changed by the CTRL.CSLPR and CHMDC[1:0] bits, use the STSL register to confirm that the mode is switched. Do not modify the CTRL.CHMDC[1:0] bits until the mode is switched.

Note 3. After the channel transitions to channel communication mode, when 11 consecutive recessive bits have been detected, communication is ready (the STSL.COMSTS flag becomes 1) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

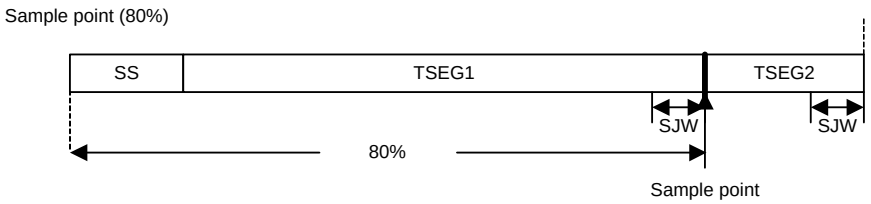
**Figure 36.16 CAN Setting Procedure after the Operation of the CAN Module is Enabled**

### 36.9.1 Clock Setting

Set the CAN clock source (fCAN) as a clock source of the CAN module. Select PCLK or CANMCLK with the GCFGL.DCS bit.

### 36.9.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments, SS, TSEG1, and TSEG2. Two of the segments, TSEG1 and TSEG2, can be set by the CFGH register for each channel. Sample point timing can be determined by setting two segments. This timing can be adjusted in units of 1 Time Quantum (referred to as Tq hereinafter). 1 Tq equals to one CAN Tq clock cycle. The CAN Tq clock is obtained by selecting the clock source with the GCFGL.DCS bit and selecting the clock division ratio with the CFGL.BRP[9:0] bits.
Figure 36.17 shows the bit timing chart. Table 36.12 shows an example of bit timing setting.




SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 16 Tq.
Set TSEG2 to a range of 2 Tq to 8 Tq.
Set SJW to a range of 1 Tq to 4 Tq.
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 25 Tq.
TSEG1 > TSEG2 ≥ SJW

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space.
  Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice of the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (resynchronization jump width)
  The SJW is a length to extend or reduce time segment to compensate an error in phase due to phase error.

**Figure 36.17 Bit Timing Chart**

**Table 36.12 Example of Bit Timing Setting**

| 1 Bit | Set Value (Tq) | | | | Sample Point (%) Note: See Figure 36.17 |
|---|---|---|---|---|---|
| | SS | TSEG1 | TSEG2 | SJW | |
| 8 Tq | 1 | 4 | 3 | 1 | 62.50 |
| | 1 | 5 | 2 | 1 | 75.00 |
| 10 Tq | 1 | 6 | 3 | 1 | 70.00 |
| | 1 | 7 | 2 | 1 | 80.00 |
| 16 Tq | 1 | 10 | 5 | 1 | 68.75 |
| | 1 | 11 | 4 | 1 | 75.00 |
| 20 Tq | 1 | 13 | 6 | 1 | 70.00 |
| | 1 | 15 | 4 | 3 | 80.00 |
| 24 Tq | 1 | 15 | 8 | 1 | 66.67 |
| | 1 | 16 | 7 | 1 | 70.83 |

## 36.9.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value (CFGL.BRP[9:0] bits), and Tq count per bit time.

Figure 36.18 shows the CAN clock control block diagram, and Table 36.13 shows an example of the communication speed setting.

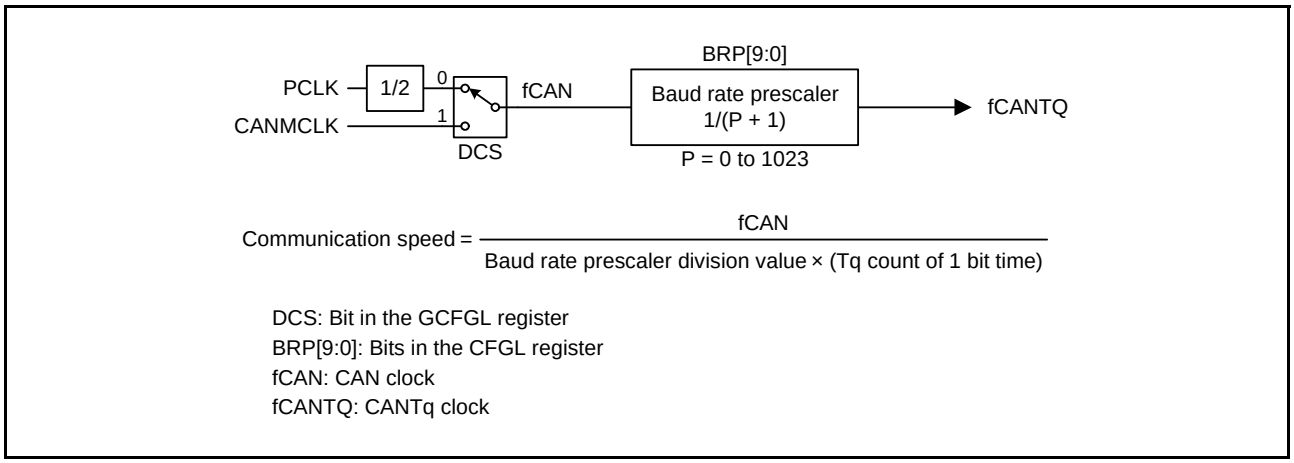



**Figure 36.18 CAN Clock Control Block Diagram**

**Table 36.13 Example of Communication Speed Setting**

| | **fCAN** | |
|---|---|---|
| **Communication Speed** | **16 MHz** | **8 MHz** |
| 1 Mbps | 8 Tq (2)<br>16 Tq (1) | 8 Tq (1) |
| 500 kbps | 8 Tq (4)<br>16 Tq (2) | 8 Tq (2)<br>16 Tq (1) |
| 250 kbps | 8 Tq (8)<br>16 Tq (4) | 8 Tq (4)<br>16 Tq (2) |
| 83.3 kbps | 8 Tq (24)<br>16 Tq (12) | 8 Tq (12)<br>16 Tq (6) |
| 33.3 kbps | 8 Tq (60)<br>10 Tq (48)<br>16 Tq (30)<br>20 Tq (24) | 8 Tq (30)<br>10 Tq (24)<br>16 Tq (15)<br>20 Tq (12) |

Note: Values in ( ) are baud rate prescaler division values.

### 36.9.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.
Up to 16 receive rules can be registered.
Figure 36.19 shows the receive rule setting procedure.

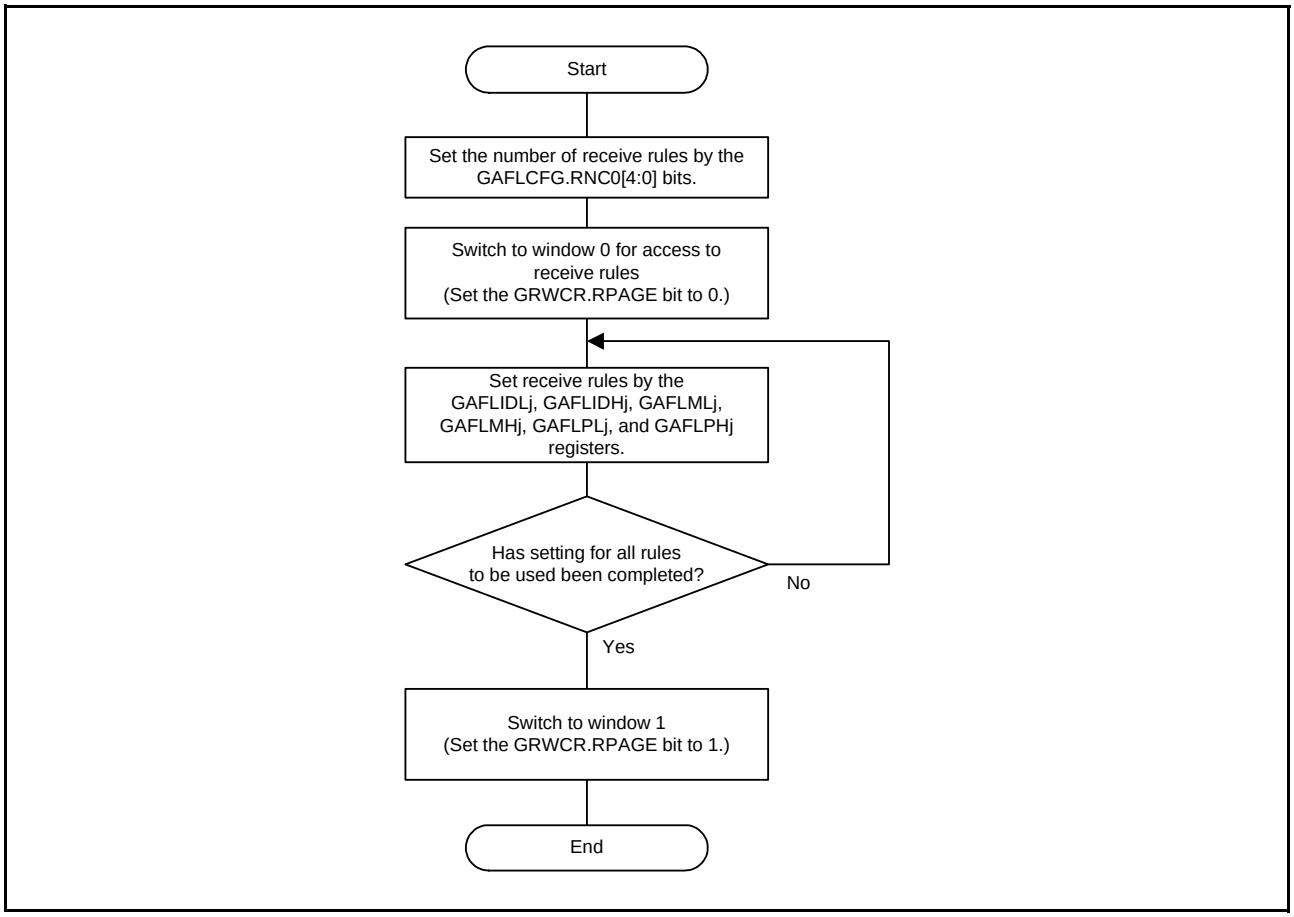



**Figure 36.19 Receive Rule Setting Procedure**

## 36.9.5 Buffer Setting

Set sizes and interrupt sources of buffers. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

Figure 36.20 shows the buffer configuration. Figure 36.21 shows the buffer setting procedure.

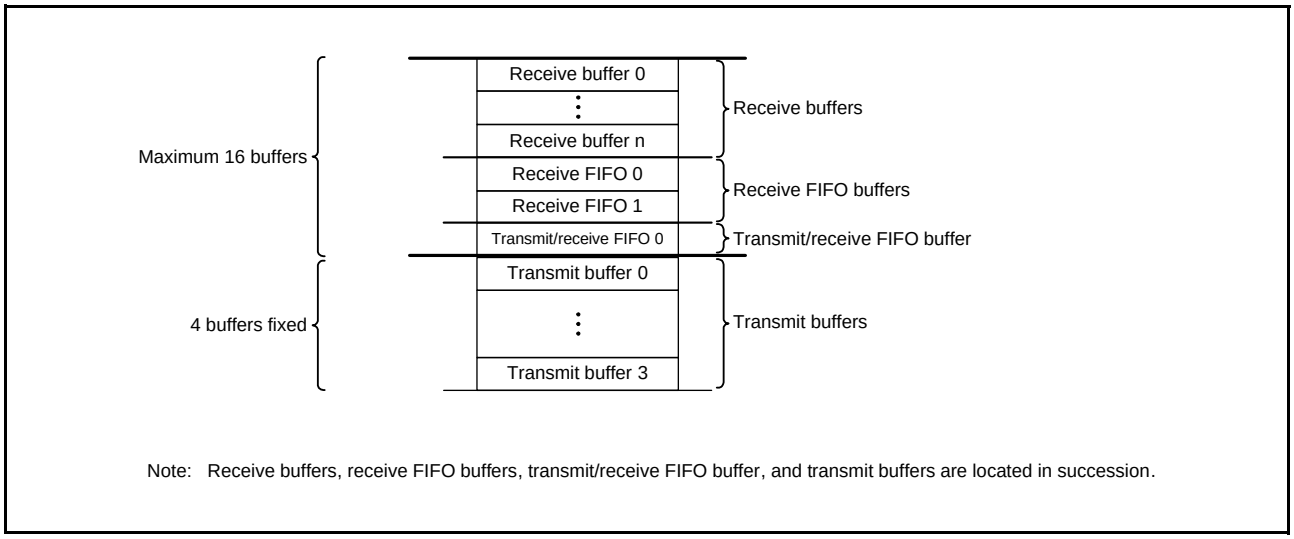



**Figure 36.20 Buffer Configuration**

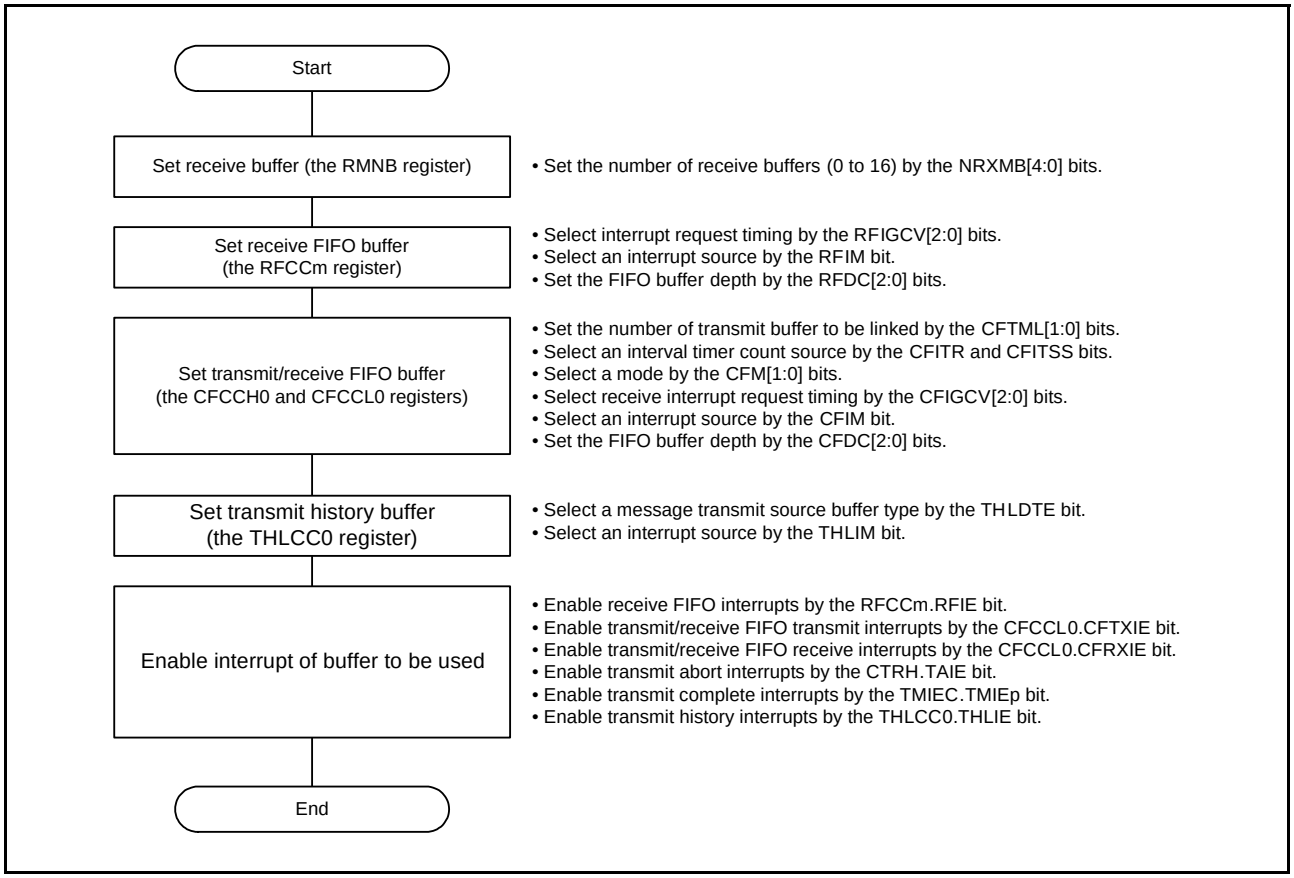



**Figure 36.21 Buffer Setting Procedure**

## 36.10 Reception Procedure

### 36.10.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMND0.RMNSn flag becomes 1 (receive buffer n contains a new message). Messages can be read from the RMIDLn, RMIDHn, RMTSn, RMPTRn, and RMDF0n to RMDF3n registers. If the next message has been received before the current message is read from the receive buffer, the message is overwritten. Figure 36.22 shows the receive buffer reading procedure.

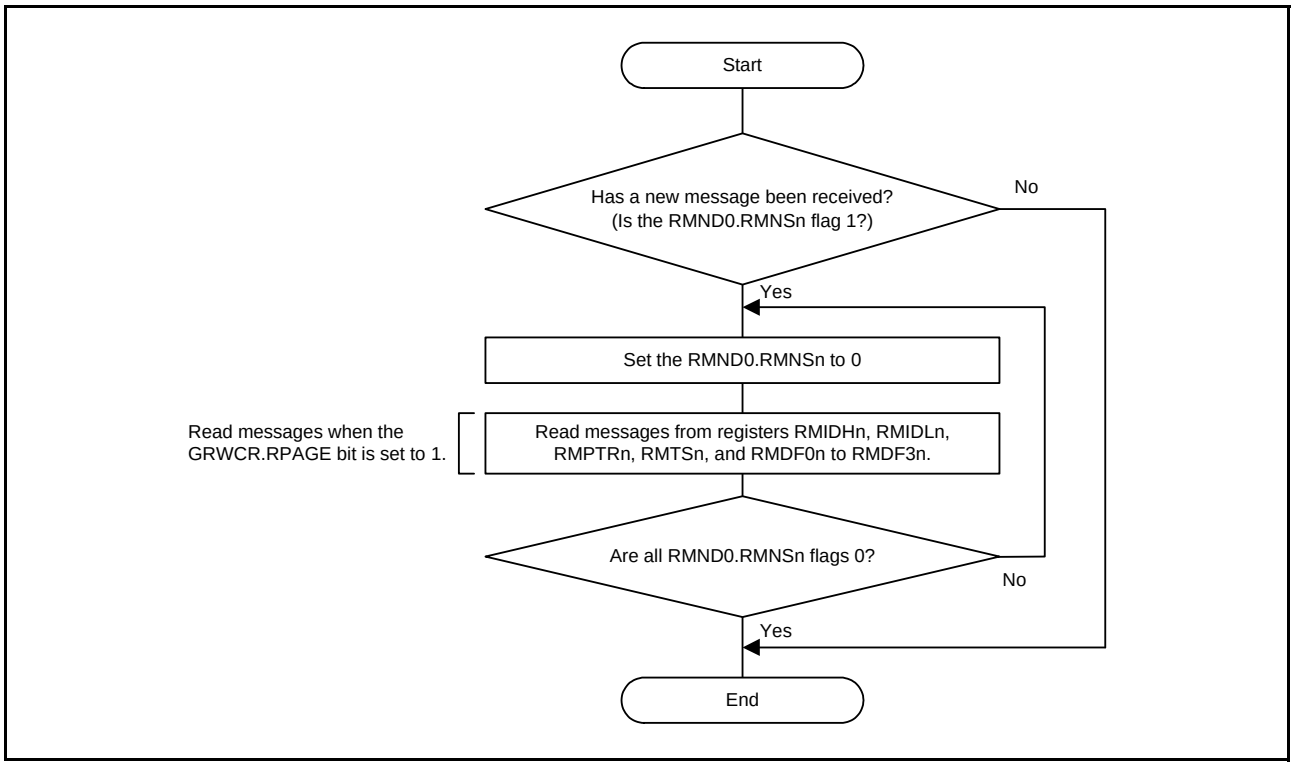



**Figure 36.22 Receive Buffer Reading Procedure**

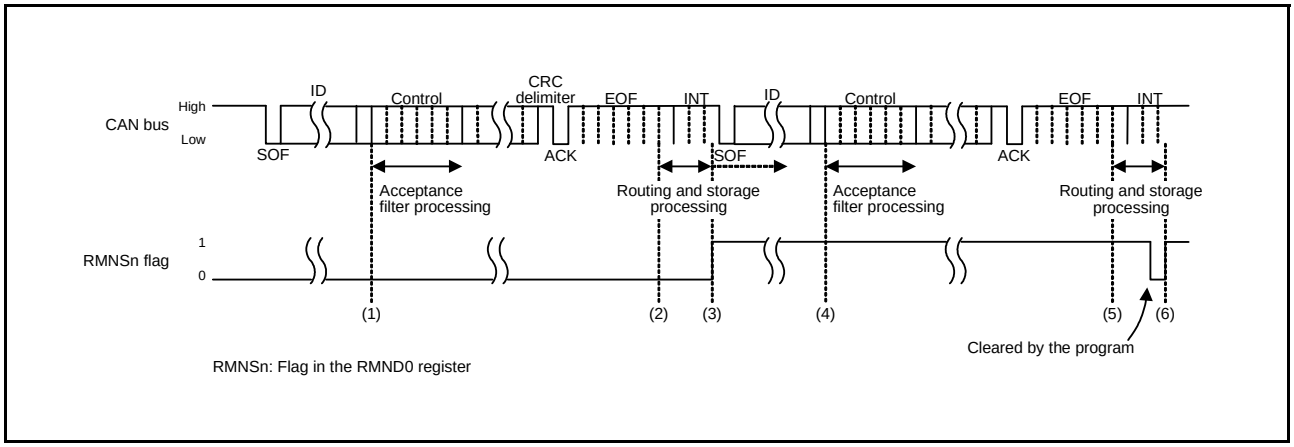



**Figure 36.23 Receive Buffer Reception Timing Chart**

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the GCFGL.DCE bit is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the corresponding RMND0.RMNSn flag becomes 1 (receive buffer contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the GCFGL.DCE bit is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(6) When the corresponding RMND0.RMNSn flag becomes 0 (receive buffer contains no new message), this flag becomes 1 again when the message storage processing starts. Even if the RMND0.RMNSn flag remains 1, a new message is overwritten to the receive buffer. The RMND0.RMNSn flag should not be set to 0 during storage of messages.

## 36.10.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode, the corresponding message count display counter (RFSTSm.RFMC[5:0] flags or CFSTS0.CFMC[5:0] flags) is incremented. At this time, when the RFCCm.RFIE bit (receive FIFO interrupt is enabled) or the CFCCL0.CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) is set to 1, an interrupt request is generated. Received messages can be read from the RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m registers (receive FIFO buffers) or the CFIDL0, CFIDH0, CFTS0, CFPTR0, and CFDF00 to CFDF30 registers (transmit/receive FIFO buffers). Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.
When the message count display counter value matches the FIFO buffer depth (a value set by the RFCCm.RFDC[2:0] bits or the CFCCL0.CFDC[2:0] bits), the RFFLL or CFFLL flag becomes 1 (the receive FIFO buffer is full).
When all messages have been read out of the FIFO buffer, the RFSTSm.RFEMP flag or CFSTS0.CFEMP flag becomes 1 (the receive FIFO buffer contains no unread message (buffer empty)).
If the RFCCm.RFE bit or the CFCCL0.CFE bit is set to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFSTSm.RFIF flag or CFSTS0.CFRXIF flag) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically set to 0. Set the interrupt request flag to 0 by the program.

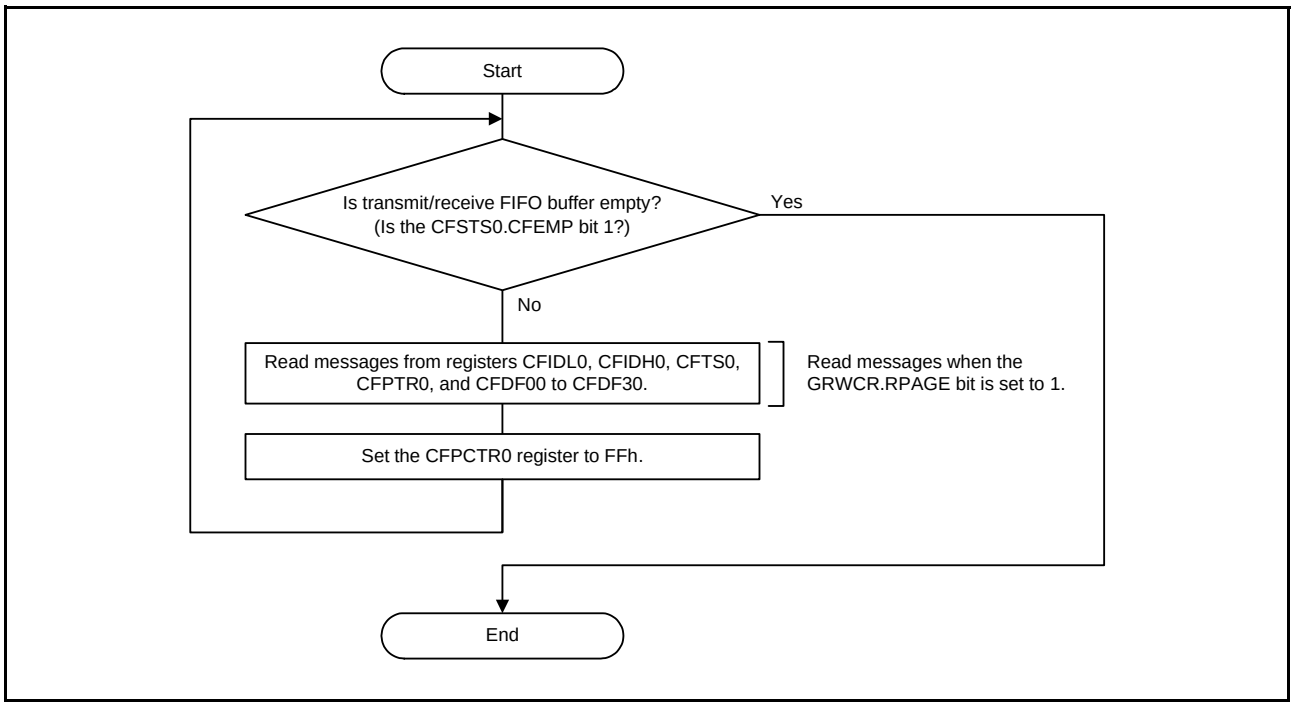



**Figure 36.24 Transmit/Receive FIFO Buffer Reading Procedure**

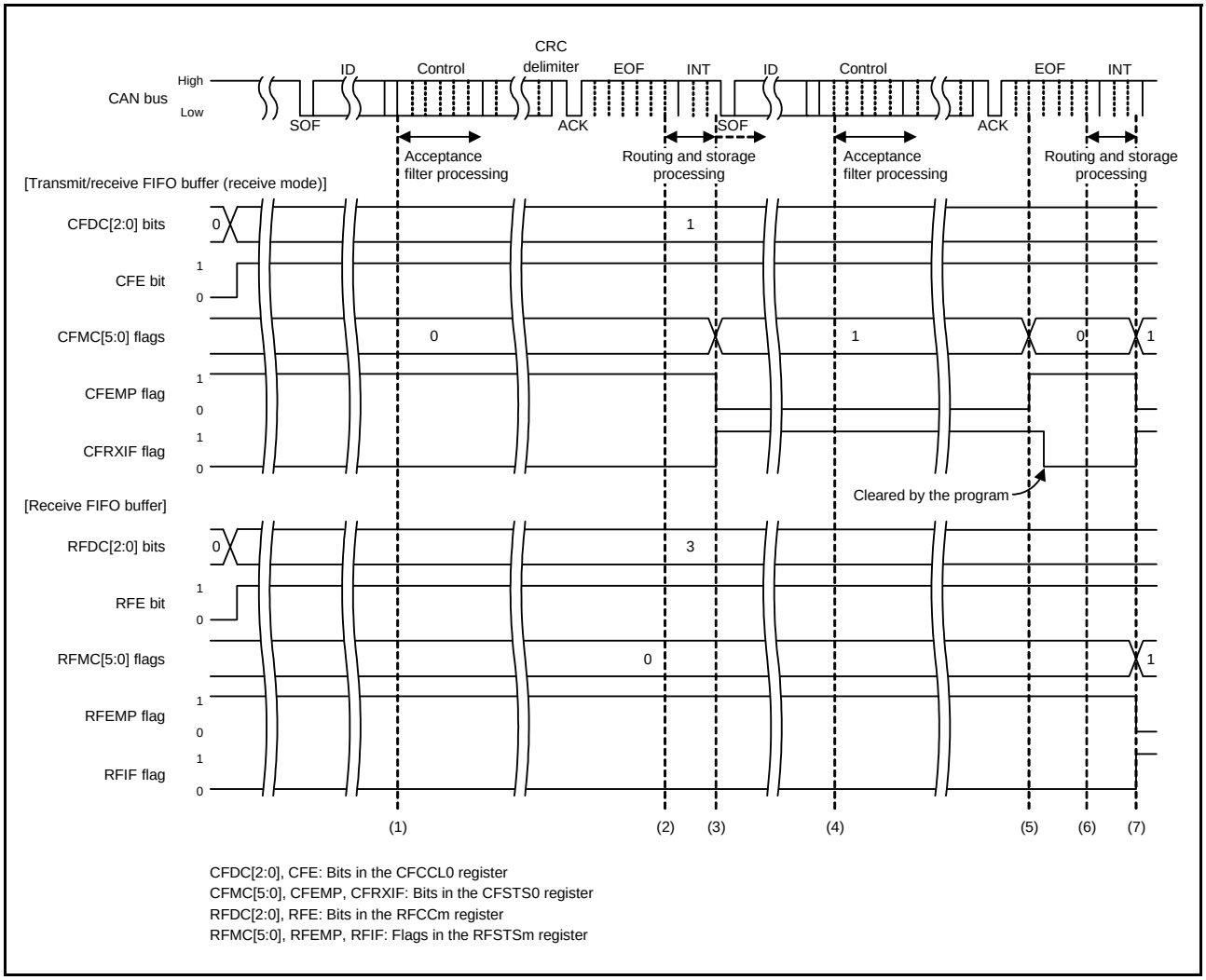



**Figure 36.25 FIFO Buffer Reception Timing Chart**

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the GCFGL.DCE bit is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing and the CFCCL0.CFE value is 1 (transmit/receive FIFO buffers are used) and the CFCCL0.CFDC[2:0] value is 001b or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFSTS0.CFMC[5:0] value is incremented and becomes 01h. When the CFCCL0.CFIM bit is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFSTS0.CFRXIF flag becomes 1 (a transmit/receive FIFO receive interrupt request is present). The CFSTS0.CFRXIF flag can be reset to 0 by the program.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) Read received messages from the CFIDL0, CFIDH0, CFTS0, CFPTR0, and CFDF00 to CFDF30 registers and write FFh to the CFPCTR0 register. Thereby the CFSTS0.CFMC[5:0] flags are decremented and become 00h, and the CFSTS0.CFEMP flag becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

(6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the GCFGL.DCE bit is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(7) The message is stored in the transmit/receive FIFO buffer set in receive mode, when the message has passed through the DLC filter process if the CFCCL0.CFE bit is set to 1 (transmit/receive FIFO buffers are used) and the

CFCCL0.CFDC[2:0] bits are set to 001b or more.
The CFSTS0.CFMC[5:0] value is incremented to 01h. When the CFCCL0.CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFSTS0.CFRXIF flag becomes 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer, if the RFCCm.RFE bit is set to 1 (receive FIFO buffers are used) and RFCCm.RFDC[2:0] bits are set to 001b or more. The RFSTSm.RFMC[5:0] value is incremented to 01h. When the RFCCm.RFIM bit is set to 1 (an interrupt occurs each time a message has been received), the RFSTSm.RFIF flag becomes 1 (a receive FIFO interrupt request is present).

## 36.11 Transmission Procedure

### 36.11.1 Procedure for Transmission from Transmit Buffers

Figure 36.26 shows the procedure for transmission from transmit buffers.
Figure 36.27 shows a timing chart where messages are transmitted from two transmit buffers and transmission has been successfully completed. Figure 36.28 shows a timing chart where messages are transmitted from two transmit buffers and transmit abort has been completed.

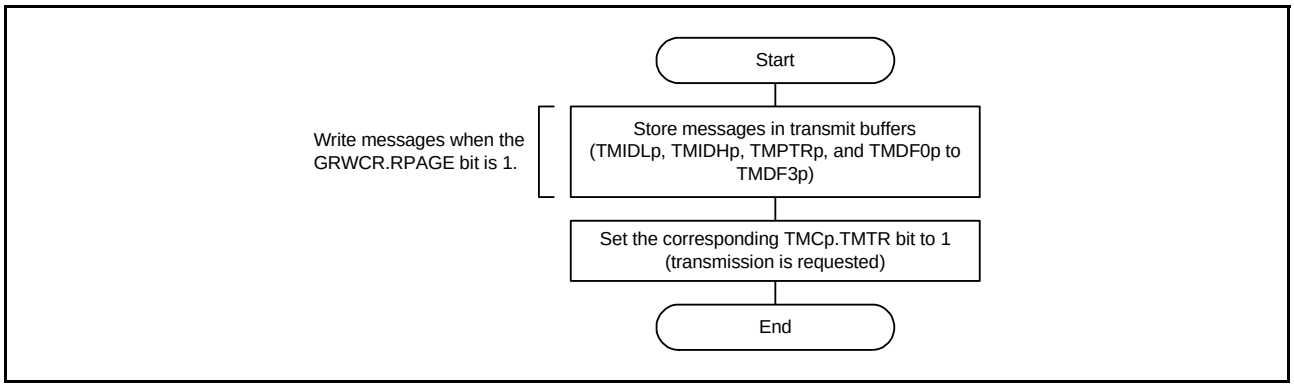



**Figure 36.26 Procedure for Transmission from Transmit Buffers**

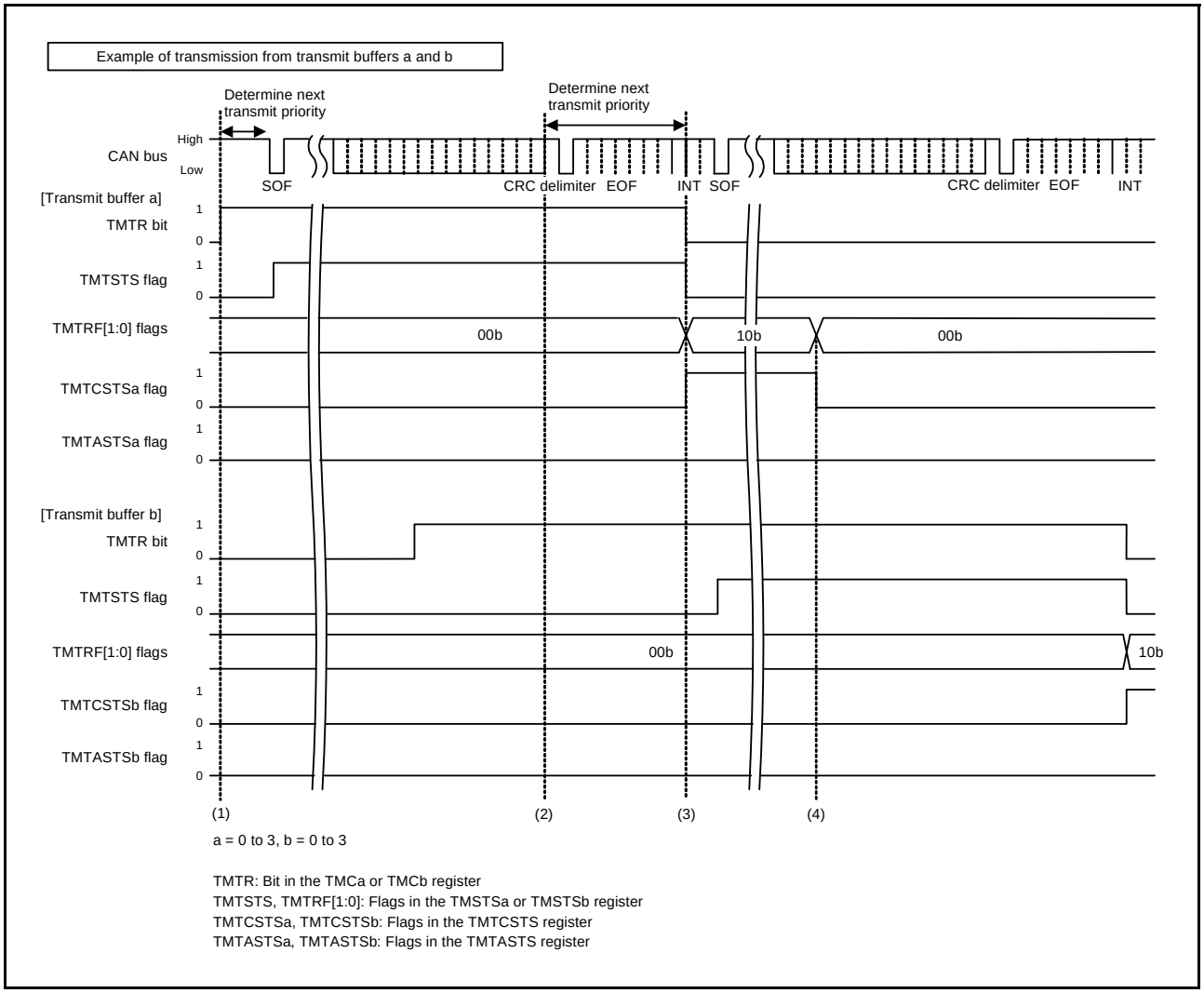



**Figure 36.27 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)**

(1) When the TMCa.TMTR bit (a = 0 to 3) is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the corresponding TMSTSa.TMTSTS flag becomes 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission.

(3) When transmit completes successfully, the TMSTSa.TMTRF[1:0] flags become 10b (transmission has been completed (without transmit abort request)), the TMSTSa.TMTSTS flag and the TMCa.TMTR bit become 0, and the TMTCSTS.TMTCSTSa flag becomes 1. When the TMIEC.TMIEa value is 1 (transmit buffer interrupt is enabled), a transmit interrupt request is generated. To clear the interrupt request, set the TMSTSa.TMTRF[1:0] flags to 00b (transmission is in progress or no transmit request is present).

(4) Before starting the next transmission, set the TMSTSa.TMTRF[1:0] flags to 00b. Write the next message to the transmit buffer, and then set the TMCa.TMTR bit to 1 (transmission is requested). The TMCa.TMTR bit can be set to 1 only when the TMSTSa.TMTRF[1:0] flag value is 00b.
If an arbitration lost has occurred after transmission is started, the TMSTSa.TMTSTS flag becomes 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration lost, the priority determination processing is reexecuted during transmission of an error frame.

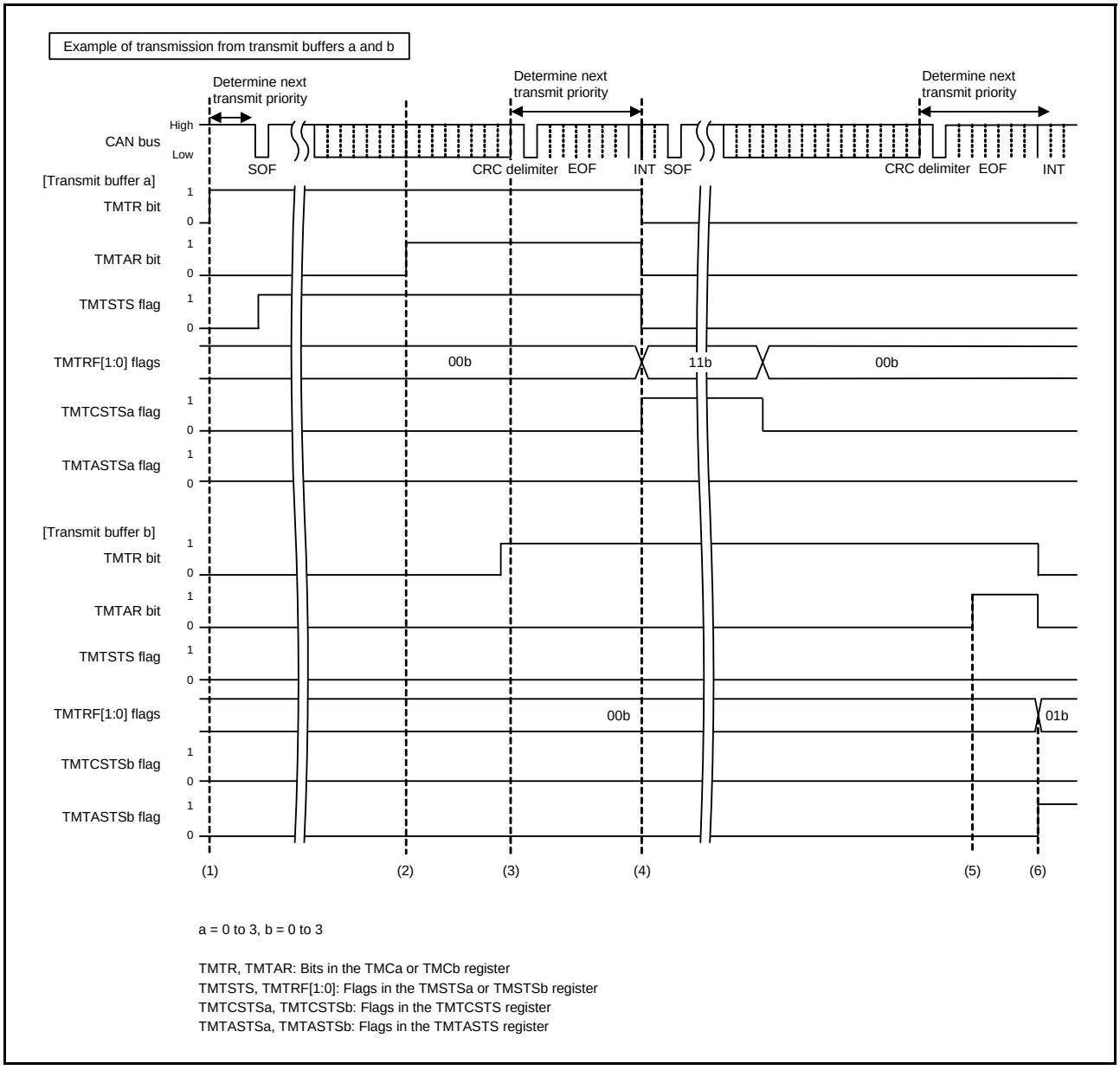



**Figure 36.28 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)**

(1) When the TMCa.TMTR bit (a = 0 to 3) is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the corresponding TMSTSa.TMTSTS flag becomes 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration lost occurs even if the TMCa.TMTAR bit is set to 1 (transmit abort is requested).

(3) The priority determination starts with the CRC delimiter for the next transmission. In this timing chart, buffer b is not selected as the next transmit buffer.

(4) When transmit completes successfully, the TMSTSa.TMTRF[1:0] flags become 11b (transmission has been completed (with transmit abort request)), the TMSTSa.TMTSTS flag and the TMCa.TMTR bit become 0, and the TMTCSTS.TMTCSTSa flag becomes 1.
When the TMIEC.TMIEa value is 1 (transmit buffer interrupt is enabled), a transmit interrupt request is generated.

To clear the interrupt request, set the TMSTSa.TMTRF[1:0] flags to 00b (transmission is in progress or no transmit request is present).

(5) While another CAN node is transmitting data on the CAN bus (TMSTSa.TMTSTS flag = 0), if the TMCa.TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMCa.TMTR bit cannot be set to 0.

(6) After the internal processing time has passed, the transmission is terminated and the TMSTSa.TMTRF[1:0] flags become 01b and the TMTASTS.TMTASTSa flag becomes 1. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMSTSa.TMTRF flag becomes 01b. At this time, the TMCa.TMTR and TMTAR bits become 0.
When transmit abort is completed with the CTRH.TAIE bit set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMSTSa.TMTRF[1:0] flags to 00b.

If an arbitration lost has occurred after the CAN channel started transmission, the TMSTSa.TMTSTS flag becomes 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration lost, the priority determination processing is reexecuted during transmission of an error frame.

### 36.11.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 36.29 shows the procedure for transmission from transmit/receive FIFO buffers.
Figure 36.30 shows a timing chart where messages are transmitted from the transmit/receive FIFO buffers and transmission has been successfully completed. Figure 36.31 shows a timing chart where messages are transmitted from the transmit/receive FIFO buffers and transmit abort has been completed.

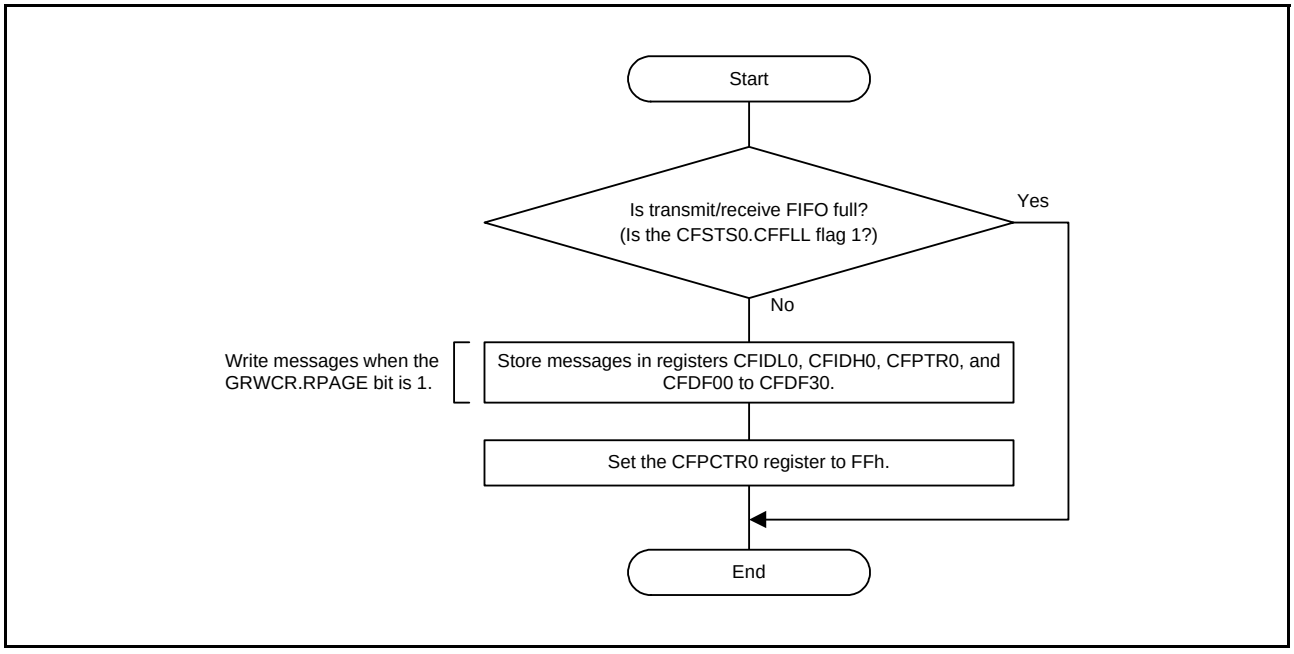



**Figure 36.29 Procedure for Transmission from Transmit/Receive FIFO Buffers**

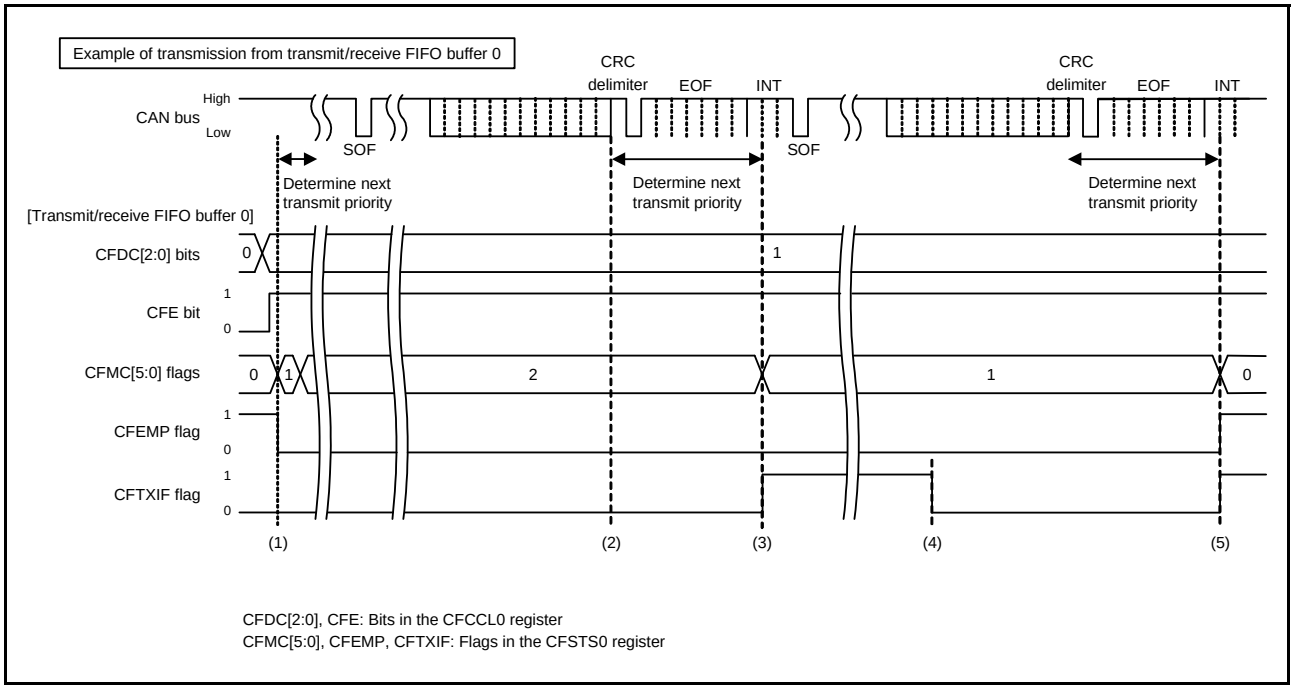



**Figure 36.30 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)**

(1) While the CAN bus is idle, when the CFCCL0.CFE value is 1 (transmit/receive FIFO buffer 0 is used) and the CFCCL0.CFDC[2:0] value is 001b (4 messages) or more and the CFSTS0.CFMC[5:0] value is 01h or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission.

(3) When transmit completes successfully, the CFSTS0.CFMC[5:0] value is decremented. Setting the CFCCL0.CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFSTS0.CFTXIF flag to 1 (a transmit/receive FIFO transmit interrupt request is present).

(4) The CFSTS0.CFTXIF flag can be cleared by the program.

(5) Message transmission from transmit/receive FIFO buffer 0 has been completed and the CFSTS0.CFMC[5:0] value is decremented. The CFSTS0.CFMC[5:0] flags become 00h and therefore the CFSTS0.CFEMP flag becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).
Transmission is continued until the CFSTS0.CFEMP flag becomes 1. It is possible to continuously store transmit messages in FIFO buffers until the CFSTS0.CFFLL flag becomes 1 (the transmit/receive FIFO buffer is full).

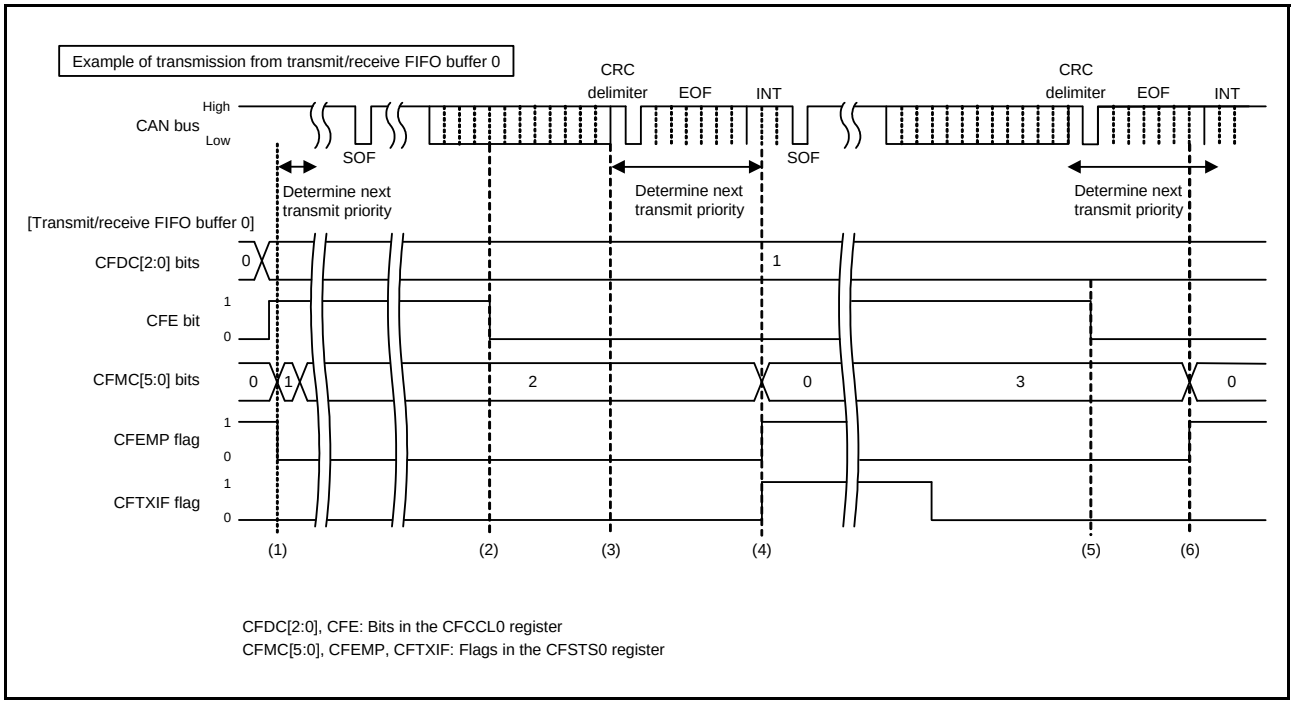



**Figure 36.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)**

(1) While the CAN bus is idle, when the CFCCL0.CFE value is 1 (transmit/receive FIFO buffer 0 is used) and the CFCCL0.CFDC[2:0] value is 001b (4 messages) or more and the CFSTS0.CFMC[5:0] value is 01h or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts.

(2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration lost occurs even if the CFCCL0.CFE bit is set to 0 (no transmit/receive FIFO buffer 0 is used).

(3) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. In this figure, transmit/receive FIFO buffer 0 is not selected as a buffer for the next transmission.

(4) When transmit completes successfully, the CFSTS0.CFMC[5:0] value becomes 00h. Setting the CFCCL0.CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFSTS0.CFTXIF flag to 1 (a transmit/receive FIFO transmit interrupt request is present). The CFSTS0.CFTXIF flag can be cleared by the program.

(5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer 0), transmit/receive FIFO buffer 0 cannot be disabled immediately even if the CFCCL0.CFE bit is set to 0 (no transmit/receive FIFO buffer 0 is used) during transmit priority determination. (The CFSTS0.CFEMP flag is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)

(6) After the internal processing time has passed, transmit/receive FIFO buffers are disabled and the CFSTS0.CFMC[5:0] flags become 00h and the CFSTS0.CFEMP flag becomes 1. When the transmit/receive FIFO buffer 0 is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer 0 is immediately disabled. (The CFSTS0.CFMC[5:0] flags become 00h and the CFSTS0.CFEMP flag becomes 1.)

### 36.11.3 Transmit History Buffer Reading Procedure

Transmit history data can be read from the THLACC0 register. The next data can be accessed by writing FFh to the corresponding THLPCTR0 register after reading a set of data. Figure 36.32 shows the transmit history buffer reading procedure.

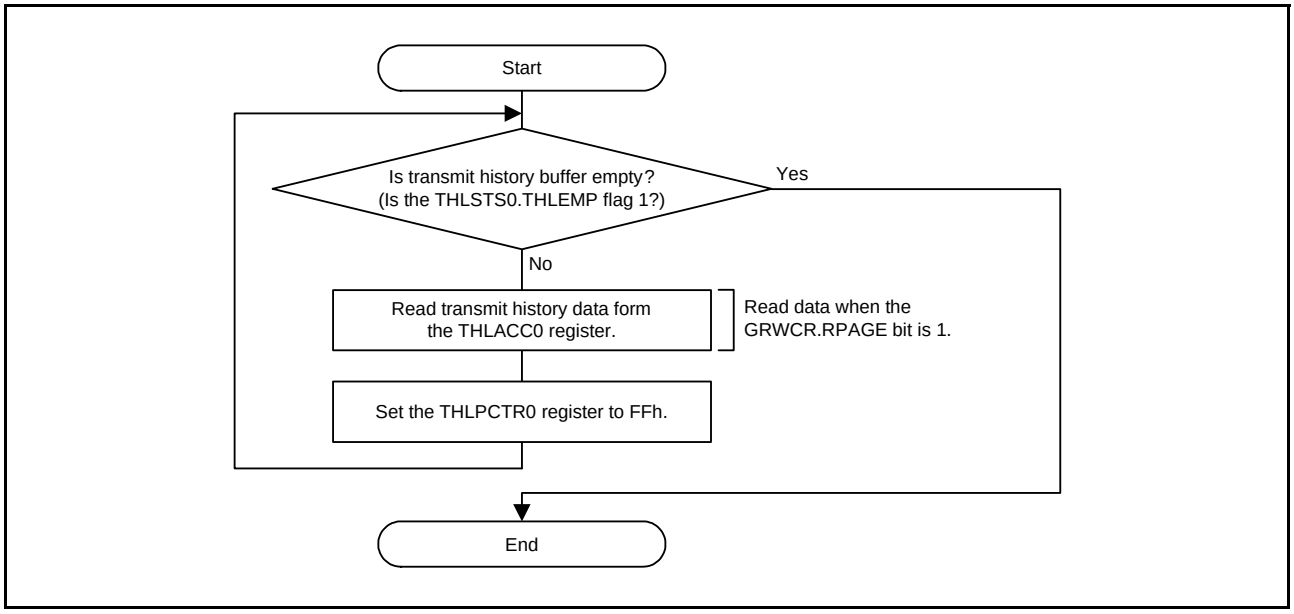



**Figure 36.32 Transmit History Buffer Reading Procedure**

## 36.12 Test Settings

### 36.12.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by receiving messages transmitted from the own node. Figure 36.33 shows the self-test mode setting procedure.

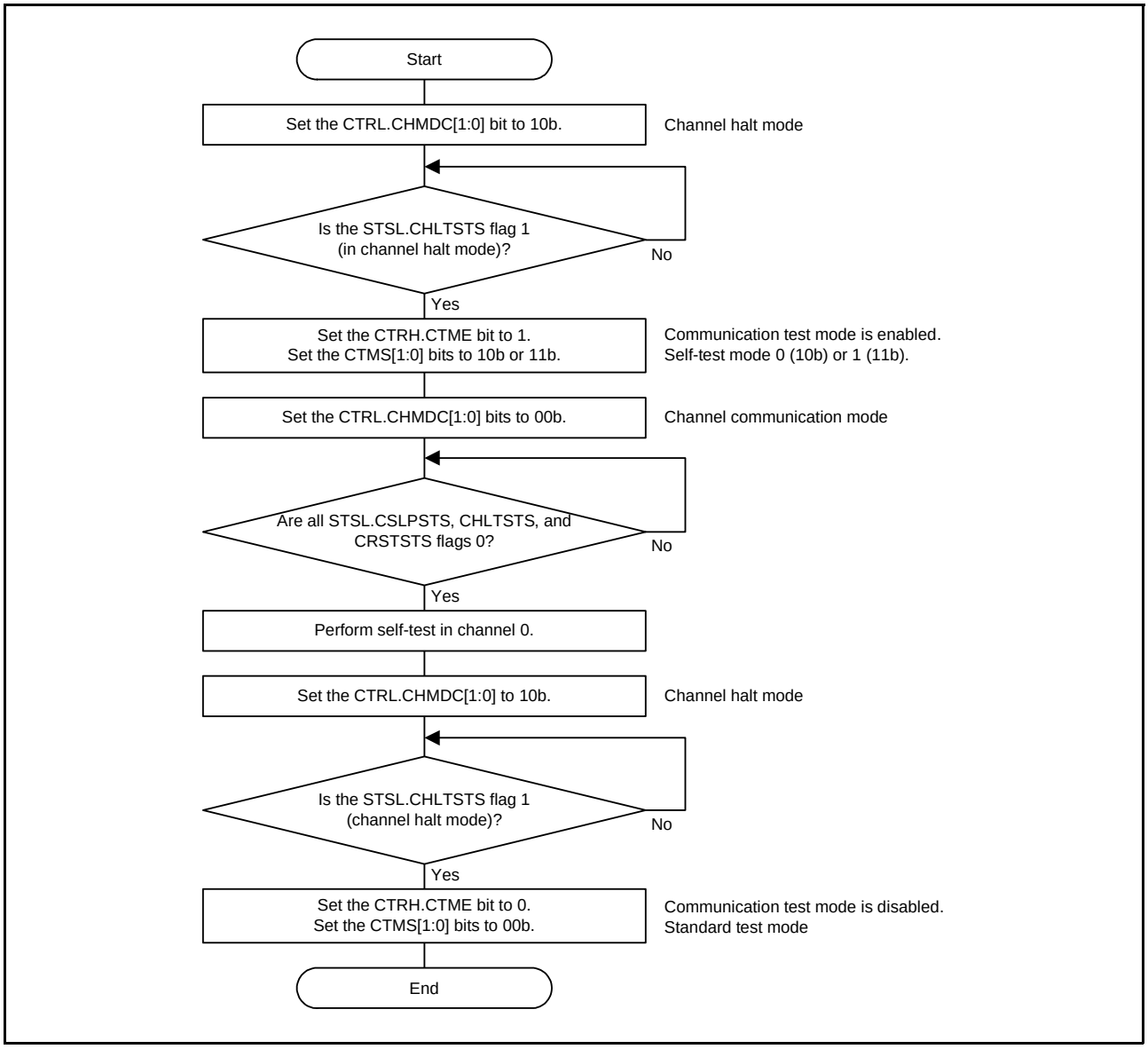



**Figure 36.33 Self-Test Mode Setting Procedure**

## 36.12.2 Protection Unlock Procedure

Since the global test functions shown in Table 36.14 are protected, write unlock data 1 and unlock data 2 in succession to the GLOCKK.LOCK[15:0] bits, and then set each test function bit to 1.

**Table 36.14 Protection Unlock Data for Test Functions**

| Test Function | Protection Unlock Data 1 | Protection Unlock Data 2 | Target Bit |
|---|---|---|---|
| RAM test | 7575h | 8A8Ah | GTSTCTRL.RTME bit |

If an incorrect value has been written to the GLOCKK.LOCK[15:0] bits, retry the procedure above from writing of unlock data 1.
Figure 36.34 shows the protection unlock procedure.

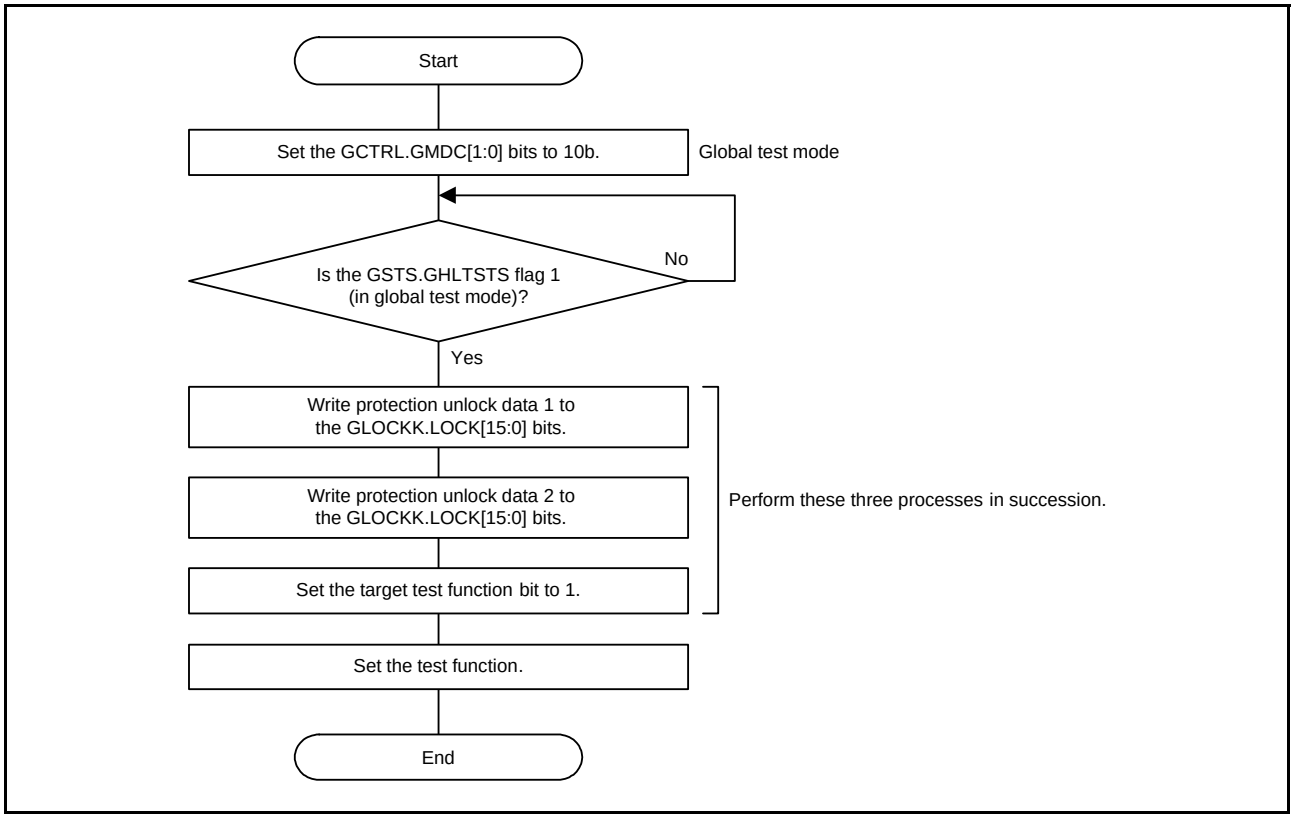



**Figure 36.34 Protection Unlock Procedure**

## 36.12.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before closing the RAM test, write 0000h to all pages of the CAN RAM.

Figure 36.35 shows the RAM test setting procedure.

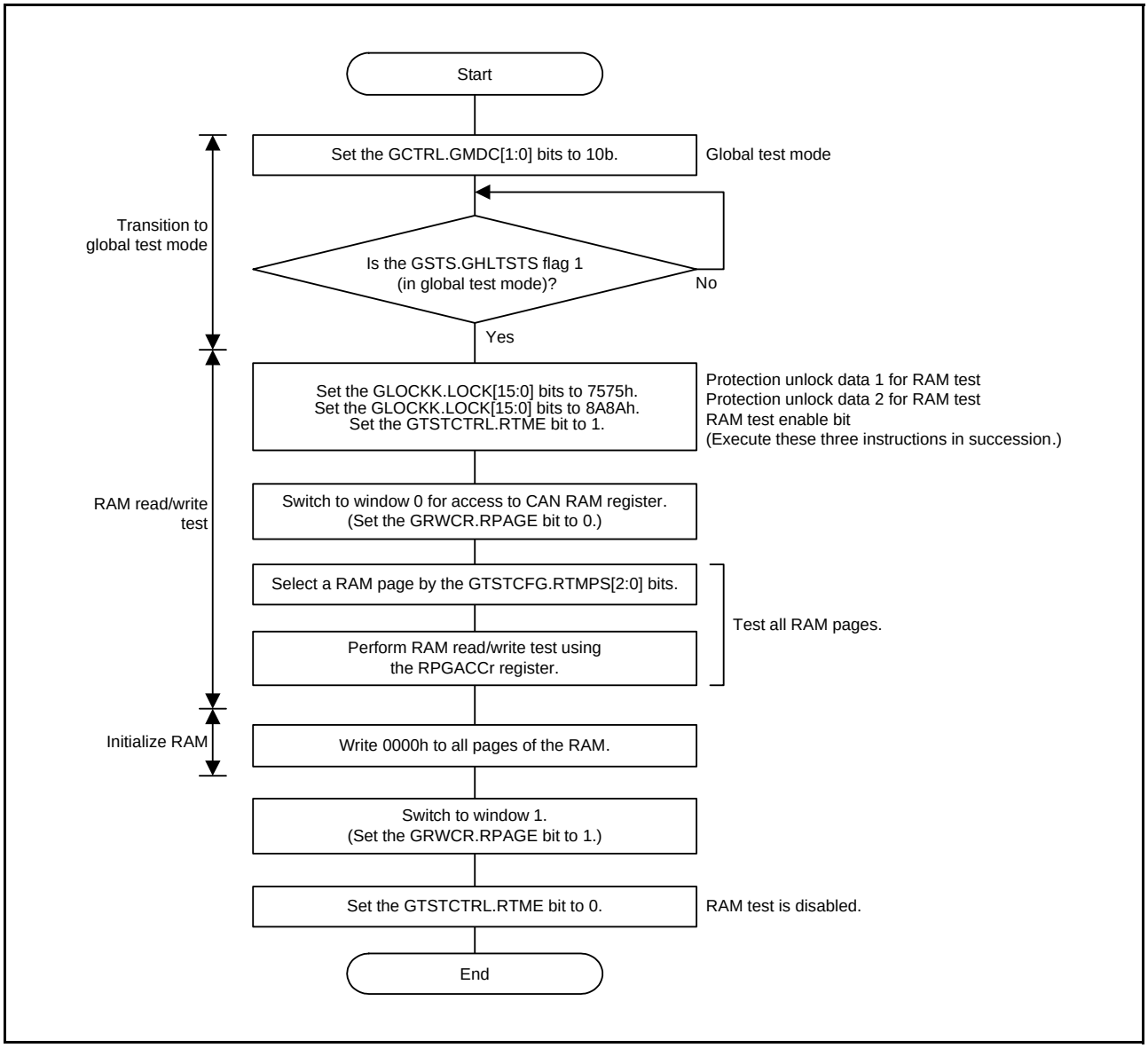



**Figure 36.35 RAM Test Setting Procedure**

## 36.13 Notes on the CAN Module

- When changing a global mode, check the GSTS.GSLPSTS, GHLTSTS, and GRSTSTS flags for transitions. When changing a channel mode, check the STSL.CSLPSTS, CHLTSTS, and CRSTSTS flags for transitions.
- The acceptance filter processing checks receive rules sequentially in ascending order from the minimum rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the minimum number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers, set the control register (TCMp) of the corresponding transmit buffer to 00h. The status register (TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers TMTRSTS, TMTCSTS, and TMTASTS), which correspond to transmit buffers linked to transmit/receive FIFO buffers remain unchanged. Set the enable bit in the corresponding interrupt enable register (the TMIEC register) to 0 (transmit buffer interrupt is disabled).
- When the CAN bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a new receive message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer, check that the transmit/receive FIFO buffer is not full.
- Since an interrupt request flag in the CAN module is not automatically set to 0 when an interrupt is accepted, the flags must be set to 0 by software. After the corresponding interrupt request flag has been set to 1, an interrupt is not generated even if an interrupt source condition is satisfied.
- In order to generate the CAN related interrupt that several interrupt sources are gathered, the following condition should be met:
  All interrupt request flags corresponding to these interrupt sources in the CAN module are set to 0 (note that this only applies to those interrupt request flags for which the corresponding interrupt enable bits shown in Table 36.11 are set to 1).
- The values of unused receive buffer registers (RMIDLn, RMIDHn, RMTSn, RMPTRn, and RMDF0n to RMDF3n (n = 0 to 15)), receive FIFO access registers (RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m (m = 0, 1)), and transmit/receive FIFO access registers (CFIDL0, CFIDH0, CFTS0, CFPTR0, and CFDF00 to CFDF30) become undefined once the CAN module exits from global reset mode and enters global operating mode or global test mode.

# 37. Serial Sound Interface (SSI)

This MCU integrates one channel of the serial sound interface (SSI) compliant with the I$^2$S bus specification. The SSI supports I$^2$S data format and MSB-first and left-justified/right-justified formats, so it can be used to send or receive audio data with various devices.

## 37.1 Overview

**Table 37.1 SSI Specifications**

| Item | Specifications |
|---|---|
| Number of channels | One channel (SSI0) |
| Operating mode | Non-compressed mode |
| Transmission formats | • I$^2$S format supported<br>• MSB-first supported<br>Left-justified/right-justified formats selectable |
| Function | • Serves as both a transmitter and a receiver<br>Channel 0 supports full-duplex communications.<br>• Capable of various audio formats<br>• SSISCK0 (serial bit clock) can be selected from among 16, 32, 48, and 64 fs (fs: Sampling rate)<br>• The master clock (MCLK) can be selected from either of the following:<br>Master clock pin for audio (AUDIO_MCLK): 1 to 25 MHz<br>Main clock<br>• Includes 8-stage FIFO buffers in transmitter and receiver<br>• Capable of selecting whether to stop word select (SSIWS0) or not when data transmission is stopped |
| Interrupt sources | Three sources<br>• Communication error<br>Transmit underflow, transmit overflow, receive underflow, receive overflow, and idle<br>• Receive data full<br>• Transmit data empty |
| Low power consumption function | Module stop state can be set. |

Figure 37.1 shows a block diagram of SSI (SSI0).

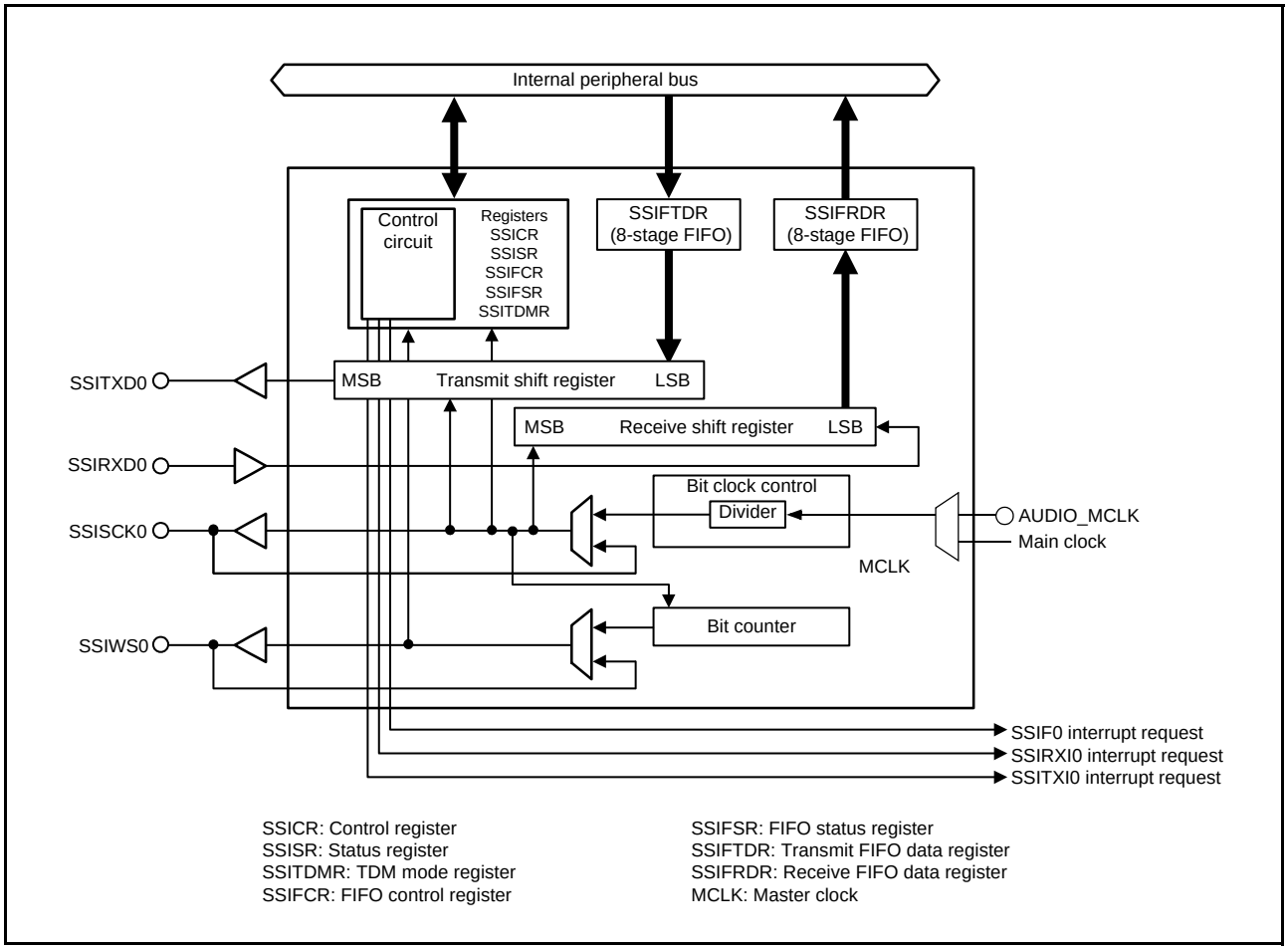



**Figure 37.1 Block Diagram of SSI (SSI0)**

Table 37.2 lists the I/O pins of the SSI.

**Table 37.2 SSI I/O Pins**

| Channel | Pin Name | I/O | Description |
|---|---|---|---|
| SSI0 | SSISCK0 | I/O | Serial bit clock pin |
| | SSIWS0 | I/O | Word selection pin |
| | SSITXD0 | Output | Serial data output pin |
| | SSIRXD0 | Input | Serial data input pin |
| | AUDIO_MCLK | Input | Master clock for audio pin (input master clock) |

## 37.2 Register Descriptions

### 37.2.1 Control Register (SSICR)

Address(es): SSI0.SSICR 0008 A500h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | CKS | TUIEN | TOIEN | RUIEN | ROIEN | IIEN | — | CHNL[1:0] | | DWL[2:0] | | | SWL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SCKD | SWSD | SCKP | SWSP | SPDP | SDTA | PDTA | DEL | CKDV[3:0] | | | | MUEN | — | TEN | REN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | REN | Receive Enable | 0: Disables receive operation.<br>1: Enables receive operation. | R/W |
| b1 | TEN | Transmit Enable | 0: Disables transmit operation.<br>1: Enables transmit operation. | R/W |
| b2 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b3 | MUEN | Mute Enable*1 | 0: Not muted.<br>1: Muted. | R/W |
| b7 to b4 | CKDV[3:0] | Serial Bit Clock Frequency Setting*3 | b7 b6 b5 b4<br>0 0 0 0: MCLK<br>0 0 0 1: MCLK/2<br>0 0 1 0: MCLK/4<br>0 0 1 1: MCLK/8<br>0 1 0 0: MCLK/16<br>0 1 0 1: MCLK/32<br>0 1 1 0: MCLK/64<br>0 1 1 1: MCLK/128<br>1 0 0 0: MCLK/6<br>1 0 0 1: MCLK/12<br>1 0 1 0: MCLK/24<br>1 0 1 1: MCLK/48<br>1 1 0 0: MCLK/96<br>Settings other than above are prohibited. | R/W |
| b8 | DEL | Serial Data Delay*3 | 0: I2S format compatibility<br>One clock cycle delay between SSIWS0 and SSITXD0/ SSIRXD0<br>1: MSB-first left-justified/right-justified format compatibility<br>No delay between SSIWS0 and SSITXD0/SSIRXD0 | R/W |
| b9 | PDTA | Parallel Data Allocation*3 | When data word length is 8 or 16 bits:<br>0: The lower bits of parallel data (SSIFTDR, SSIFRDR) are transferred prior to the upper bits.<br>1: The upper bits of parallel data (SSIFTDR, SSIFRDR) are transferred prior to the lower bits.<br><br>When data word length is 18, 20, 22, or 24 bits:<br>0: Parallel data (SSIFTDR, SSIFRDR) is left-justified.<br>1: Parallel data (SSIFTDR, SSIFRDR) is right-justified. | R/W |
| b10 | SDTA | Serial Data Alignment*3 | 0: Transmitting and receiving in the order of serial data and padding bits<br>1: Transmitting and receiving in the order of padding bits and serial data | R/W |
| b11 | SPDP | Serial Padding Polarity*3 | 0: Padding data is 0.<br>1: Padding data is 1. | R/W |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b12 | SWSP | Word Select Polarity | 0: SSIWS0 is low for the 1st system word, high for the 2nd system word.<br>1: SSIWS0 is high for the 1st system word, low for the 2nd system word. | R/W |
| b13 | SCKP | Serial Bit Clock Polarity*3 | 0: SSIWS0 and SSITXD0 change at the SSISCK0 falling edge (SSIWS0 and SSIRXD0 are sampled at the SSISCK0 rising edge).<br>1: SSIWS0 and SSITXD0 change at the SSISCK0 rising edge (SSIWS0 and SSITXD0 are sampled at the SSISCK0 falling edge). | R/W |
| b14 | SWSD | Word Select Direction*2, *3 | 0: SSIWS0 pin is input (slave mode).<br>1: SSIWS0 pin is output (master mode). | R/W |
| b15 | SCKD | Serial Bit Clock Direction*2, *3 | 0: SSISCK0 pin is input (slave mode).<br>1: SSISCK0 pin is output (master mode). | R/W |
| b18 to b16 | SWL[2:0] | System Word Length*3 | Set the system word length to the bit clock frequency/2 fs.<br>b18 b16<br>0 0 0: 8 bits (serial bit clock frequency = 16 fs)<br>0 0 1: 6 bits (serial bit clock frequency = 32 fs)<br>0 1 0: 24 bits (serial bit clock frequency = 48 fs)<br>0 1 1: 32 bits (serial bit clock frequency = 64 fs)<br>Settings other than above are prohibited. | R/W |
| b21 to b19 | DWL[2:0] | Data Word Length*3 | b21 b19<br>0 0 0: 8 bits<br>0 0 1: 16 bits<br>0 1 0: 18 bits<br>0 1 1: 20 bits<br>1 0 0: 22 bits<br>1 0 1: 24 bits<br>Settings other than above are prohibited. | R/W |
| b23, b22 | CHNL[1:0] | Channels*3 | b23 b22<br>0 0: One channel<br>Settings other than above are prohibited. | R/W |
| b24 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b25 | IIEN | Idle Interrupt Enable | 0: Disables an idle interrupt.<br>1: Enables an idle interrupt. | R/W |
| b26 | ROIEN | Receive Overflow Interrupt Enable | 0: Disables a receive overflow interrupt.<br>1: Enables a receive overflow interrupt. | R/W |
| b27 | RUIEN | Receive Underflow Interrupt Enable | 0: Disables a receive underflow interrupt.<br>1: Enables a receive underflow interrupt. | R/W |
| b28 | TOIEN | Transmit Overflow Interrupt Enable | 0: Disables a transmit overflow interrupt.<br>1: Enables a transmit overflow interrupt. | R/W |
| b29 | TUIEN | Transmit Underflow Interrupt Enable | 0: Disables a transmit underflow interrupt.<br>1: Enables a transmit underflow interrupt. | R/W |
| b30 | CKS | Audio Clock Select*3 | 0: AUDIO_MCLK input<br>1: Main clock | R/W |
| b31 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note 1. While this module is muted, low is transmitted regardless of the value of serial data, but data transmission is not stopped. Since the number of data in the transmit FIFO decreases, write dummy data to the SSIFTDR register to prevent the generation of a transmit underflow. When the MUEN bit is set to 1, the SSITXD0 pin immediately becomes low without synchronizing SSIWS0 pin.

Note 2. Set the SCKD and SWSD bits to the same value. Other settings are prohibited.

Note 3. Rewriting is allowed only in the idle state.

### REN Bit (Receive Enable)

This bit enables or disables receive operation. Setting this bit to 1 starts receive operation.

### TEN Bit (Transmit Enable)

This bit enables or disables transmit operation. Setting this bit to 1 starts transmit operation.
SSITXD0 pin of SSI0 is set as output while SSITXD0 is selected by the multi-function pin controller (MPC), regardless of the TEN bit setting.

**Table 37.3 SSITXD0 and SSIRXD0 Pin States**

| Register Settings | | | SSI0 | |
|---|---|---|---|---|
| MPC setting | TEN | REN | SSITXD0 | SSIRXD0 |
| SSI function | 0 | 0 | Output | Input |
| | 0 | 1 | Output | Input |
| | 1 | 0 | Output | Input |
| | 1 | 1 | Output | Input |
| Other than SSI function | x | x | Depends on the selected function | Depends on the selected function |

x: Don't care
—: Settings prohibited.

### CKDV[3:0] Bits (Serial Bit Clock Frequency Setting)

These bits select the frequency of the serial bit clock in master mode. Since the input clock from the SSISCK0 pin is used in slave mode, the setting of these bits is ignored. The serial bit clock is used as the operating clock of the shift register.

Calculation Example:
When fs (sampling rate) = the SSIWS0 frequency = 96 kHz and the system word length = 32 bits
The bit clock frequency = 96 kHz × 32 bits × 2 = 6.144 MHz is necessary, so set CKDV[3:0] = 0001b (MCLK/2) when MCLK = 12.288 MHz.

### PDTA Bit (Parallel Data Allocation)

The setting of this bit specifies the allocation of data to be stored in the SSIFRDR register in receive mode and the SSIFTDR register in transmit mode.
During receive operation, the SSI stores the data received from the serial bus in the SSIFRDR register according to the PDTA bit setting.
During transmit operation, the SSI stores the data stored in the SSIFTDR register in the transmit shift register, and transmits the data to the serial bus according to the PDTA bit setting.

(1) When PDTA bit is 0

| DWL[2:0] Bits | SSIFTDR and SSIFRDR Registers |
|---|---|
| 000b | 31 24 23 16 15 8 7 0<br>4th word \| 3rd word \| 2nd word \| 1st word |
| 001b | 31 16 15 0<br>2nd word \| 1st word |
| 010b | 31 14 13 0<br>Valid \| Invalid |
| 011b | 31 12 11 0<br>Valid \| Invalid |
| 100b | 31 10 9 0<br>Valid \| Invalid |
| 101b | 31 8 7 0<br>Valid \| Invalid |

(2) When PDTA bit is 1

| DWL[2:0] Bits | SSIFTDR and SSIFRDR Registers |
|---|---|
| 000b | 31 24 23 16 15 8 7 0<br>1st word \| 2nd word \| 3rd word \| 4th word |
| 001b | 31 16 15 0<br>1st word \| 2nd word |
| 010b | 31 18 17 0<br>Invalid \| Valid |
| 011b | 31 20 19 0<br>Invalid \| Valid |
| 100b | 31 22 21 0<br>Invalid \| Valid |
| 101b | 31 24 23 0<br>Invalid \| Valid |

### SCKP Bit (Serial Bit Clock Polarity)

This bit is used to select the polarity of SSISCK0 signal.

Table 37.4 lists the setting of SCKP bit and signal I/O timing.

**Table 37.4 Setting of SCKP Bit and Signal Timing**

| | SCKP Bit = 0 | SCKP Bit = 1 |
|---|---|---|
| SSIRXD0 input sampling timing for reception | SSISCK0 rising edge | SSISCK0 falling edge |
| SSITXD0 output changing timing for transmission | SSISCK0 falling edge | SSISCK0 rising edge |
| SSIWS0 input sampling timing in slave mode (SWSD bit = 0) | SSISCK0 rising edge | SSISCK0 falling edge |
| SSIWS0 output changing timing in master mode (SWSD bit = 1) | SSISCK0 falling edge | SSISCK0 rising edge |

### CHNL[1:0] Bits (Channels)

These bits select the number of channels to be decoded in each system word. Set these bits to 00b in this module.

## 37.2.2 Status Register (SSISR)

Address(es): SSI0.SSISR 0008 A504h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | TUIRQ | TOIRQ | RUIRQ | ROIRQ | IIRQ | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | TCHNO[1:0] | | TSWNO | RCHNO[1:0] | | RSWNO | IDST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IDST | Idle Status Flag | 0: SSI communication is in progress.<br>1: SSI communication is idle. | R |
| b1 | RSWNO | Receive System Word Number Flag | Receive word number | R |
| b3, b2 | RCHNO[1:0] | Receive Channel Number Flag | These bits are read as 00b. | R |
| b4 | TSWNO | Transmit System Word Number Flag | Transmit word number | R |
| b6, b5 | TCHNO[1:0] | Transmit Channel Number Flag | These bits are read as 00b. | R |
| b24 to b7 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b25 | IIRQ | Idle Interrupt Status Flag | 0: Not in idle state<br>1: In idle state | R |
| b26 | ROIRQ | Receive Overflow Interrupt Status Flag | 0: No receive overflow has occurred.<br>1: A receive overflow has occurred. | R/(W) *1 |
| b27 | RUIRQ | Receive Underflow Interrupt Status Flag | 0: No receive underflow has occurred.<br>1: A receive underflow has occurred. | R/(W) *1 |
| b28 | TOIRQ | Transmit Overflow Interrupt Status Flag | 0: No transmit overflow has occurred.<br>1: A transmit overflow has occurred. | R/(W) *1 |
| b29 | TUIRQ | Transmit Underflow Interrupt Status Flag | 0: No transmit underflow has occurred.<br>1: A transmit underflow has occurred. | R/(W) *1 |
| b31, b30 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Writing 0 after confirming the flag to be 1 clears the flag. To clear flags, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to a status flag indicating 0.

### IDST Flag (Idle Status Flag)

This status flag indicates that the SSI is in idle state where communication is stopped.

This flag is set to 0 when communication starts after the SSICR.TEN bit or SSICR.REN bit is set to 1. Also, this flag is set to 1 if both the TEN and REN bits are set to 0 and system word communication is completed.

If the external device stops inputting the serial bit clock before communication is completed, this flag is not set to 1.

### RSWNO Flag (Receive System Word Number Flag)

The initial value of this flag is 1, and its value is inverted when the data is transferred from the receive shift register to the SSIFRDR register.

This flag is initialized to 1 when the SSICR.REN bit value is changed from 0 to 1.

When the data word length specified by the SSICR.DWL[2:0] bits is 18 bits or more, this flag indicates which system word the data in the SSIFRDR register represents.

#### TSWNO Flag (Transmit System Word Number Flag)

This status flag indicates the current word number.
The initial value of this is 1, and its value is inverted when the data is transferred from the SSIFTDR register to the transmit shift register.
This flag is initialized to 1 when the SSICR.TEN bit value is changed from 0 to 1.
When the data word length specified by the SSICR.DWL[2:0] bits is 18 bits or more, this flag indicates the system word that is in the data transferred from the SSIFTDR register to the transmit shift register.

#### IIRQ Flag (Idle Interrupt Status Flag)

This status flag indicates whether this module is in idle state.
This flag is set regardless of the value of the SSICR.IIEN bit to allow polling.
The interrupt can be masked by setting the SSICR.IIEN bit to 0, but the interrupt cannot be cleared by writing 0 to this flag.
If IIRQ flag = 1 and SSICR.IIEN bit = 1, an interrupt occurs.

#### ROIRQ Flag (Receive Overflow Interrupt Status Flag)

This status flag indicates that receive data was supplied at a higher rate than was required. If a receive overflow occurs, stop reception and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.ROIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If ROIRQ flag = 1 and SSICR.ROIEN bit = 1, an interrupt occurs.
If ROIRQ flag = 1, the data was transferred from the transmit shift register to the SSIFRDR register while the receive FIFO is full (SSIFSR.RDC[3:0] flags = 8h). This may lead to the loss of data.

Note: When an overflow occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSI interface.

#### RUIRQ Flag (Receive Underflow Interrupt Status Flag)

This status flag indicates that receive data was supplied at a lower rate than was required. If a receive underflow occurs, stop reception and start the flowchart again from the beginning.
This flag is set to 1 regardless of the setting of the SSICR.RUIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If RUIRQ flag = 1 and SSICR.RUIEN bit = 1, an interrupt occurs.
If RUIRQ flag = 1, the SSIFRDR register was read while the receive FIFO is empty (SSIFSR.RDC[3:0] flags = 0h). This may cause invalid receive data to be stored.

#### TOIRQ Flag (Transmit Overflow Interrupt Status Flag)

This status flag indicates that transmit data was supplied at a higher rate than was required. If a transmit overflow occurs, stop transmission and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.TOIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If TOIRQ flag = 1 and SSICR.TOIEN bit = 1, an interrupt occurs.
If TOIRQ flag = 1, the SSIFTDR register had data written to it while the transmit FIFO is full (SSIFSR.TDC[3:0] flags = 8h). This may lead to the loss of data.

#### TUIRQ Flag (Transmit Underflow Interrupt Status Flag)

This status flag indicates that transmit data was supplied at a lower rate than was required. If a transmit underflow occurs, stop transmission and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.TUIEN bit. This flag can be set to 0 by writing 0 after

confirming it to be 1.
If TUIRQ flag = 1 and SSICR.TUIEN bit = 1, an interrupt occurs.
If TUIRQ flag = 1, the SSIFTDR register did not have data written to it before it was required for transmission. This may lead to the same data being transmitted once more.

Note: When a transmit underflow occurs, the last data input to the SSIFTDR register is transmitted until this module is in the idle state after transmission is stopped.

## 37.2.3 FIFO Control Register (SSIFCR)

Address(es): SSI0.SSIFCR 0008 A510h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUCKE | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSIRST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | TTRG[1:0] | | RTRG[1:0] | | TIE | RIE | TFRST | RFRST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RFRST | Receive FIFO Data Register Reset*4 | 0: Release the receive FIFO data reset.<br>1: Initiates the receive FIFO data reset. | R/W |
| b1 | TFRST | Transmit FIFO Data Register Reset*4 | 0: Release the transmit FIFO data reset.<br>1: Initiates the transmit FIFO data reset. | R/W |
| b2 | RIE | Receive Data Full Interrupt Enable | 0: Receive data full interrupt (RXI) request is disabled.<br>1: Receive data full interrupt (RXI) request is enabled.*1 | R/W |
| b3 | TIE | Transmit Data Empty Interrupt Enable | 0: Transmit data empty interrupt (TXI) request is disabled.<br>1: Transmit data empty interrupt (TXI) request is enabled.*2 | R/W |
| b5, b4 | RTRG[1:0] | Receive FIFO Threshold Setting *4 | b5 b4<br>0 0: 1<br>0 1: 2<br>1 0: 4<br>1 1: 6 | R/W |
| b7, b6 | TTRG[1:0] | Transmit FIFO Threshold Setting *4 | b7 b6<br>0 0: 7 (1)*3<br>0 1: 6 (2)*3<br>1 0: 4 (4)*3<br>1 1: 2 (6)*3 | R/W |
| b15 to b8 | — | Reserved | These bits are read as undefined. The write value should be 0. | R/W |
| b16 | SSIRST | SSI Software Reset | 0: Clears the SSI software reset.<br>1: Initiates the SSI software reset. | R/W |
| b30 to b17 | — | Reserved | These bits are read as undefined. The write value should be 0. | R/W |
| b31 | AUCKE | Master Clock Enable*4 | 0: The master clock is disabled.<br>1: The master clock is enabled. | R/W |

Note 1. The RXI request can be cleared by setting the SSIFSR.RDF flag to 0 (see the description of the SSIFSR.RDF flag for details) or RIE bit to 0.
Note 2. The TXI request can be cleared by setting the SSIFSR.TDE flag to 0 (see the description of the SSIFSR.TDE flag for details) or TIE bit to 0.
Note 3. The values in parenthesis are the number of empty stages in SSIFTDR at which the SSIFSR.TDE flag is set.
Note 4. Rewriting is allowed only in the idle state.

The SSIFCR register resets the number of the data bytes stored in the SSIFTDR and SSIFRDR registers, and specifies transmit FIFO and receive FIFO threshold values.

#### RFRST Bit (Receive FIFO Data Register Reset)

This bit invalidates the data in the SSIFRDR register to reset the FIFO to an empty state.

#### TFRST Bit (Transmit FIFO Data Register Reset)

This bit invalidates the data in the SSIFTDR register to reset the FIFO to an empty state.

#### RIE Bit (Receive Data Full Interrupt Enable)

This bit enables or disables generation of receive data full interrupt (RXI) requests when the SSIFSR.RDF flag is set to 1 during reception.

#### TIE Bit (Transmit Data Empty Interrupt Enable)

This bit enables or disables generation of transmit data empty interrupt (TXI) requests when the SSIFSR.TDE flag is set to 1 during transmit operation.

#### RTRG[1:0] Bits (Receive FIFO Threshold Setting)

These bits specify the receive FIFO threshold value. When the number of received data bytes stored in the SSIFRDR register (receive FIFO) has become equal to or greater than the value specified by the RTRG[1:0] bits, the SSIFSR.RDF flag is set to 1 and reading the received data is requested. If the SSIFCR.RIE bit is 1 at this time, a receive data full interrupt (RXI) request is generated.

#### TTRG[1:0] Bits (Transmit FIFO Threshold Setting)

These bits specify the transmit FIFO threshold value. When the number of transmit data bytes stored in the SSIFTDR register (transmit FIFO) has become equal to or less than the value specified by the TTRG[1:0], the SSIFSR.TDE flag is set to 1 and writing the transmit data is requested. If the SSIFCR.TIE bit is 1 at this time, a transmit data empty interrupt (TXI) request is generated.

#### SSIRST Bit (SSI Software Reset)

Writing 1 to this bit initializes the SSI internal status, registers other than the SSIFCR register, and bits other than this bit in the SSIFCR register. Since this bit is not automatically cleared to 0, confirm that 1 is written to it before writing 0. Do not write 0 to this bit and 1 to other bits at the same time. After modifying this bit, confirm that its value is modified before proceeding to the next processing.

## 37.2.4 FIFO Status Register (SSIFSR)

Address(es): SSI0.SSIFSR 0008 A514h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | TDC[3:0] | | | | — | — | — | — | — | — | — | TDE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | RDC[3:0] | | | | — | — | — | — | — | — | — | RDF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RDF | Receive Data Full Flag | 0: Number of received data bytes in the SSIFRDR register is less than the specified receive FIFO threshold value.<br>1: Number of received data bytes in the SSIFRDR register is equal to or greater than the specified receive FIFO threshold value. | R/(W) *1 |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 to b8 | RDC[3:0] | Receive Data Indicate Flag | Indicate the number of data units stored in the SSIFRDR register. | R |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b16 | TDE | Transmit Data Empty Flag | 0: Number of data bytes for transmission in the SSIFTDR register is greater than the specified transmit FIFO threshold value.<br>1: Number of data bytes for transmission in the SSIFTDR register is equal to or less than the specified transmit FIFO threshold value.*2 | R/(W) *1 |
| b23 to b17 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b27 to b24 | TDC[3:0] | Transmit Data Indicate Flag | Indicate the number of data units stored in the SSIFTDR register. | R |
| b31 to b28 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Writing 0 after confirming the flag to be 1 clears the flag. To clear flags, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to a status flag indicating 0.

Note 2. Since the SSIFTDR register is an 8-stage FIFO register, the amount of data that can be written to it while TDE flag = 1 is “8 - specified transmit FIFO threshold value” bytes at maximum. Writing more data will be ignored. The number of data bytes in the SSIFTDR register is indicated in the TDC[3:0] flags.

The SSIFSR register consists of status flags indicating the operating status of the SSIFTDR register and SSIFRDR register.

### RDF Flag (Receive Data Full Flag)

This flag indicates that, when the received data is transferred to the SSIFRDR register, the number of data bytes in the SSIFRDR register has become equal to or greater than the receive FIFO threshold value, and thus reading the received data from the SSIFRDR register has been enabled.

[Setting condition]

- The number of receive data bytes that is equal to or greater than the value specified by the SSIFCR.RTRG[1:0] bits is stored in the SSIFRDR register.

[Clearing conditions]

- 0 is written to the RDF flag after the RDF flag is confirmed to be 1.
- Received data is read from the SSIFRDR register using DMA or DTC transfer (transfer of the last block in block transfer). Do not clear the RDF flag to 0 during DMA or DTC transfer.

Note: Since the SSIFRDR register is a 32-byte FIFO register, the maximum number of data bytes that can be read from it while the RDF flag is 1 is indicated in the RDC[3:0] flags. If reading data from the SSIFRDR register is continued after all the data is read, undefined values will be read.

#### RDC[3:0] Flags (Receive Data Indicate Flag)

These flags indicate the number of data bytes stored in the SSIFRDR register.
RDC[3:0] flags = 0h indicates no received data. RDC[3:0] flags = 8h indicates that 32 bytes of received data is stored in the SSIFRDR register.

#### TDE Flag (Transmit Data Empty Flag)

This flag indicates that, when data is transferred from the SSIFTDR register to the transmit shift register, the number of data bytes in the SSIFTDR register has become less than the transmit FIFO threshold value, and thus writing transmit data to the SSIFTDR register has been enabled.
[Setting condition]

- The number of the transmit data bytes written to the SSIFTDR register is equal to or less than the value specified by the SSIFCR.TTRG[1:0] bits.

[Clearing conditions]

- 0 is written to the TDE flag after the TDE flag is confirmed to be 1.
- Transmit data is written to the SSIFTDR register using DMA or DTC transfer (transfer of the last block in block transfer). Do not clear the TDE flag to 0 during DMA or DTC transfer.

Note: Since the SSIFTDR register is a 32-byte FIFO register, the maximum number of bytes that can be written to it while the TDE flag is 1 is 8 – TDC[3:0]. If writing data to the SSIFTDR register is continued after all the data is written, writing will be invalid and an overflow occurs.

#### TDC[3:0] Flags (Transmit Data Indicate Flag)

These flags indicate the number of data bytes stored in the SSIFTDR register.
TDC[3:0] flags = 0h indicates no data for transmission. TDC[3:0] flags = 8h indicates that 32 bytes of data for transmission is stored in the SSIFTDR register.

### 37.2.5 Transmit FIFO Data Register (SSIFTDR)

Address(es): SSI0.SSIFTDR 0008 A518h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

The SSIFTDR register is a write-only FIFO register consisting of eight stages of 32-bit registers for storing transmit data. Write transmit data to the SSIFTDR register in 64-bit (two stages of FIFO) units regardless of the data word length setting. If transmit data ends on a 32-bit boundary, write 00000000h for the other 32 bits, and stop transmission while 64-bit writing is completed. When the transmit shift register is empty, the SSI transfers the transmit data written to the SSIFTDR register to start serial transmission, which can be continued until the SSIFTDR register becomes empty. Note that when the SSIFTDR register is full of data (32 bytes), the next data cannot be written to it. If writing is attempted, it will be ignored and an overflow occurs.

### 37.2.6 Receive FIFO Data Register (SSIFRDR)

Address(es): SSI0.SSIFRDR 0008 A51Ch

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

The SSIFRDR register is a read-only FIFO register consisting of eight stages of 32-bit registers for storing received data. Each time 4 bytes of serial data is received, the SSI stores the received serial data in the SSIFRDR register from the receive shift register according to the PDTA bit setting. Receive operation can be continued until a maximum 32 bytes of data have been stored to in the SSIFRDR register. The SSIFRDR register can be read but cannot be written to. Note that when the SSIFRDR register is read when it stores no received data, undefined values will be read and a receive underflow occurs.
After the SSIFRDR register becomes full of received data, the data received thereafter will be lost and a receive overflow occurs.

## 37.2.7 TDM Mode Register (SSITDMR)

Address(es): SSI0.SSITDMR 0008 A520h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CONT | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | CONT | WS Continue Mode*1 | 0: Disables WS continue mode.<br>1: Enables WS continue mode. | R/W |
| b31 to b9 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. This bit can be set only in master mode (SSICR.SCKD bit = 1 and SSICR.SWSD bit = 1).

The SSITDMR register is a readable/writable 32-bit register that enables or disables WS continue mode.

## 37.3 Operation

### 37.3.1 Bus Format

This module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode. The bus format can be selected from one of the six modes shown in Table 37.5.

**Table 37.5 Bus Format**

| | TEN | REN | SCKD | SWSD | MUEN | IIEN | TOIEN | TUIEN | ROIEN | RUIEN | CONT | SWSP | DEL | PDTA | SDTA | SPDP | SCKP | SWL[2:0] | DWL[2:0] | CHNL[1:0] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Non-compression slave receiver | 0 | 1 | 0 | 0 | Control bits | | | | | | | Configuration bits | | | | | | | | |
| Non-compression slave transmitter | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | |
| Non-compression slave transceiver | 1 | 1 | 0 | 0 | | | | | | | | | | | | | | | | |
| Non-compression master receiver | 0 | 1 | 1 | 1 | | | | | | | | | | | | | | | | |
| Non-compression master transmitter | 1 | 0 | 1 | 1 | | | | | | | | | | | | | | | | |
| Non-compression master transceiver | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | |

### 37.3.2 Non-Compressed Mode

This SSI supports non-compressed mode only. It supports the $I^2S$ compatible format as well as MSB-first and left-justified/right-justified.

#### (1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

#### (2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

#### (3) Slave Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

#### (4) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module. If the incoming data does not follow the configured format, operation is not guaranteed.

#### (5) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module.

### (6) Master Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module.

### (7) Operating Settings Related to Word Length

All bits related to the SSICR register's word length are valid in non-compressed modes. There are many configurations this module supports, but some of the combinations are shown below for the I²S compatible format, MSB-first and left-justified format, and MSB-first and right-justified format.
In this section SSITXD0 and SSIRXD0 are referred to SSIDATA.

- I²S Compatible Format

Figure 37.2 and Figure 37.3 show the I²S compatible format both without and with padding.
Padding occurs when the data word length is smaller than the system word length.

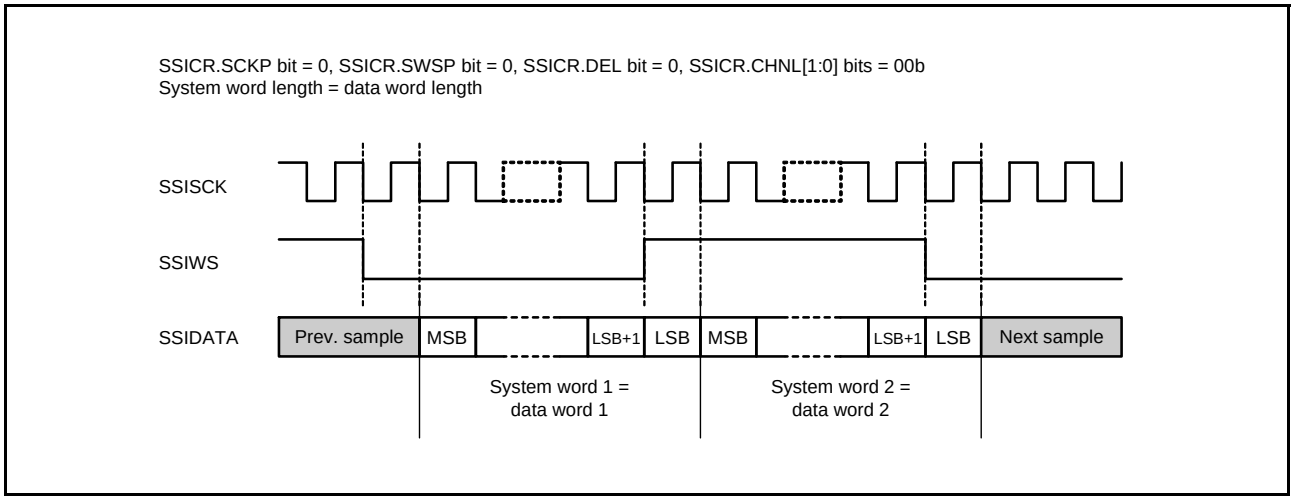



**Figure 37.2 I²S Compatible Format (without Padding)**

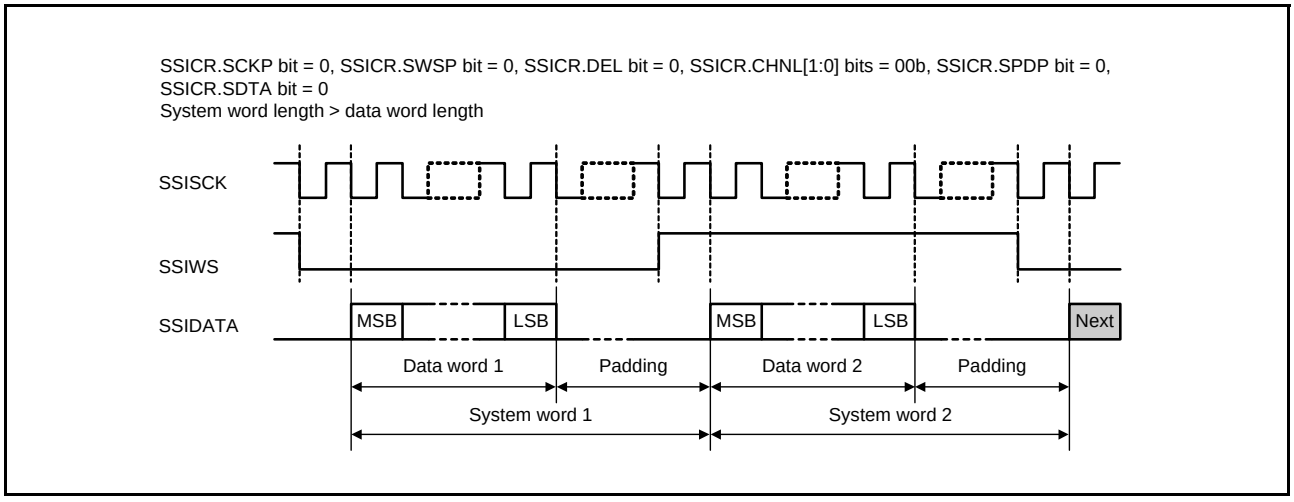



**Figure 37.3 I²S Compatible Format (with Padding)**

- MSB-First and Left-Justified Format

Figure 37.4 shows the MSB-first and left-justified format with padding.

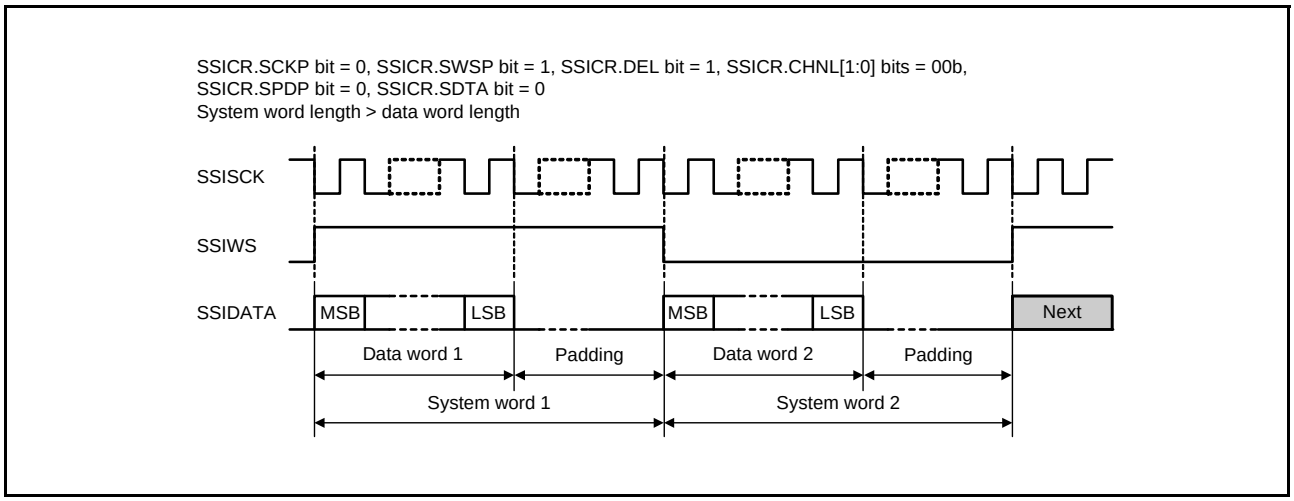



**Figure 37.4 MSB-First and Left-Justified Format (Transmitted and Received in the Order of Serial Data and Padding Bits)**

- MSB-First and Right-Justified Format

Figure 37.5 shows the MSB-first and right-justified format with padding.

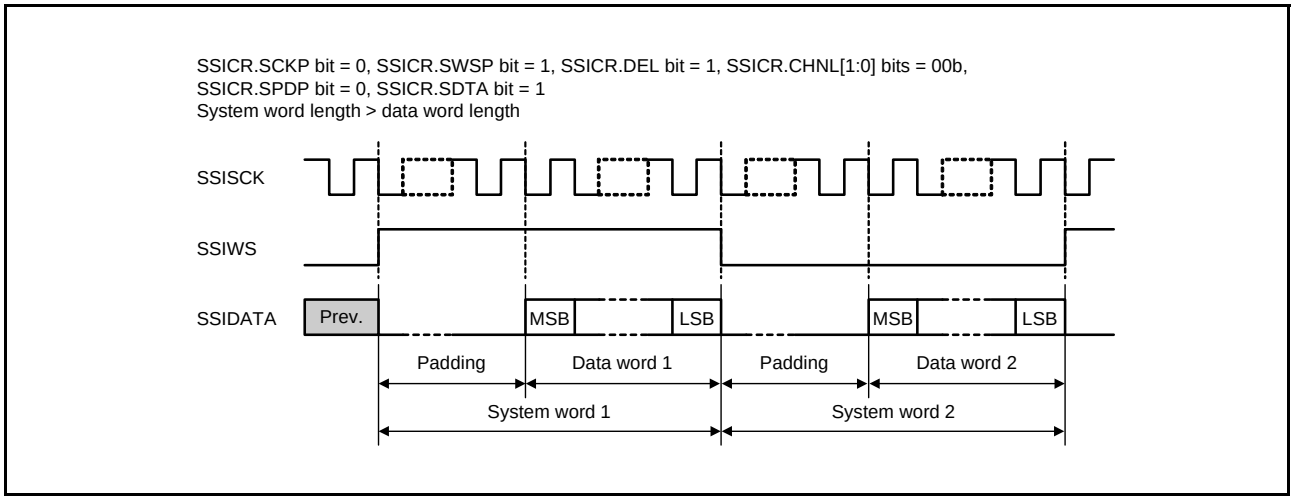



**Figure 37.5 MSB-First and Right-Justified Format (Transmitted and Received in the Order of Padding Bits and Serial Data)**

Table 37.6 shows the number of padding bits for each of the valid setting.

**Table 37.6 Number of Padding Bits per System Word for Each Valid Setting**

| SSICR.CHNL[1:0] Bits (Decoded Channels per System Word) | | SSICR.SWL[2:0] Bits (System Word Length) | | SSICR.DWL[2:0] Bits (Data Word Length) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 000b | 001b | 010b | 011b | 100b | 101b |
| | | | | 8 bits | 16 bits | 18 bits | 20 bits | 22 bits | 24 bits |
| 00b | 1 channel | 000b | 8 bits | 0 | — | — | — | — | — |
| | | 001b | 16 bits | 8 | 0 | — | — | — | — |
| | | 010b | 24 bits | 16 | 8 | 6 | 4 | 2 | 0 |
| | | 011b | 32 bits | 24 | 16 | 14 | 12 | 10 | 8 |

### (8) Operating Settings Other than Word Length Related Settings

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful.

These configuration bits are described below with reference to the basic format sample in Figure 37.6.
In Figure 37.6 to Figure 37.14, a system word length of 6 bits and a data word length of 4 bits are used for simplification of these figures.

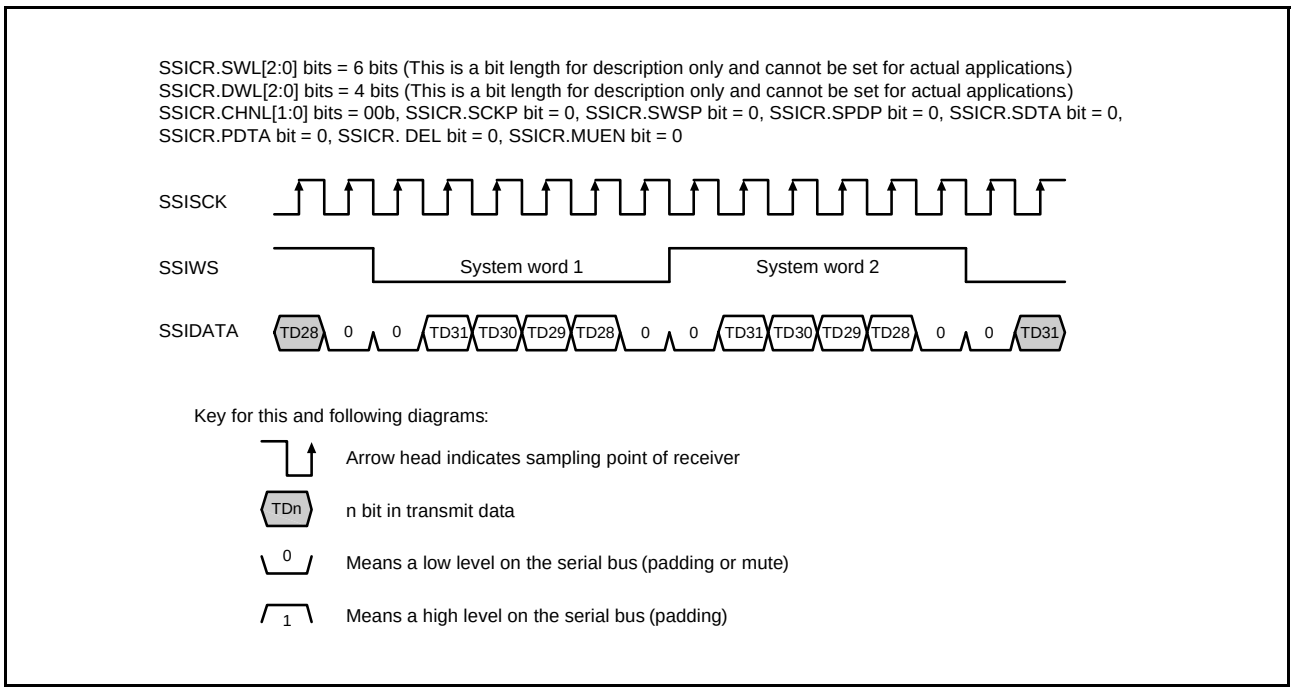



**Figure 37.6 Basic Format Sample (Transmit Mode)**

- Inverted Clock

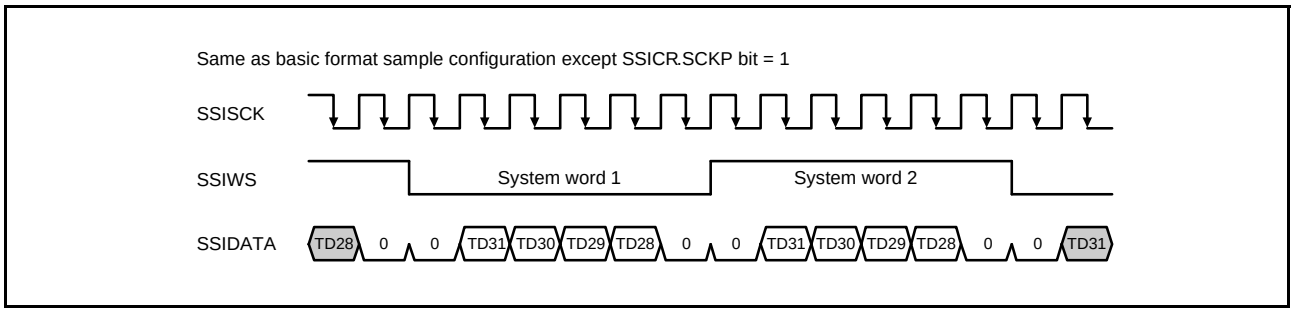



**Figure 37.7 Inverted Clock**

- Inverted Word Select

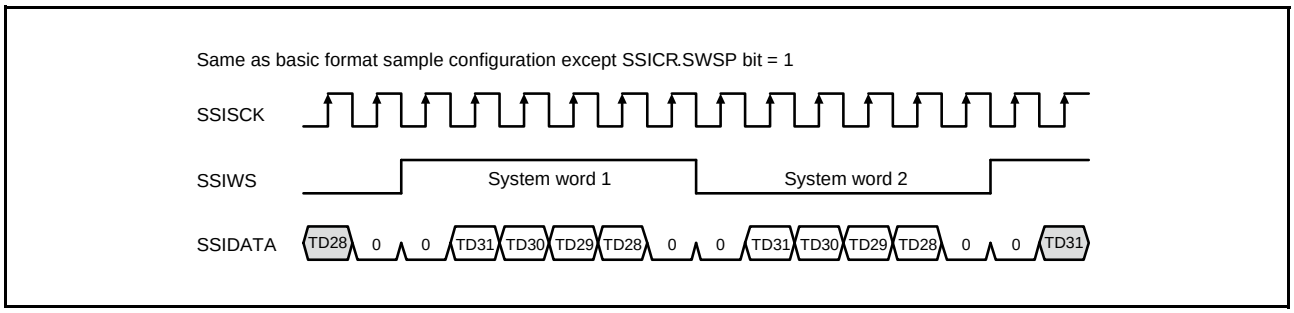



**Figure 37.8 Inverted Word Select**

- Inverted Padding Polarity

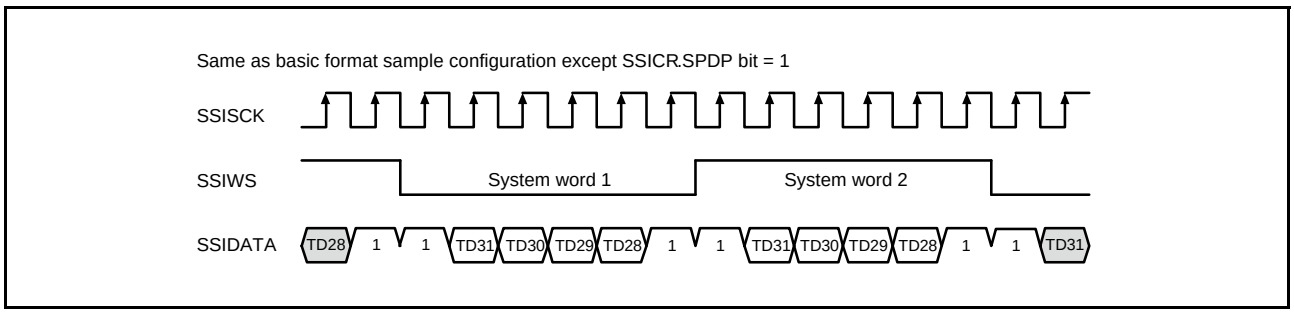



**Figure 37.9 Inverted Padding Polarity**

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

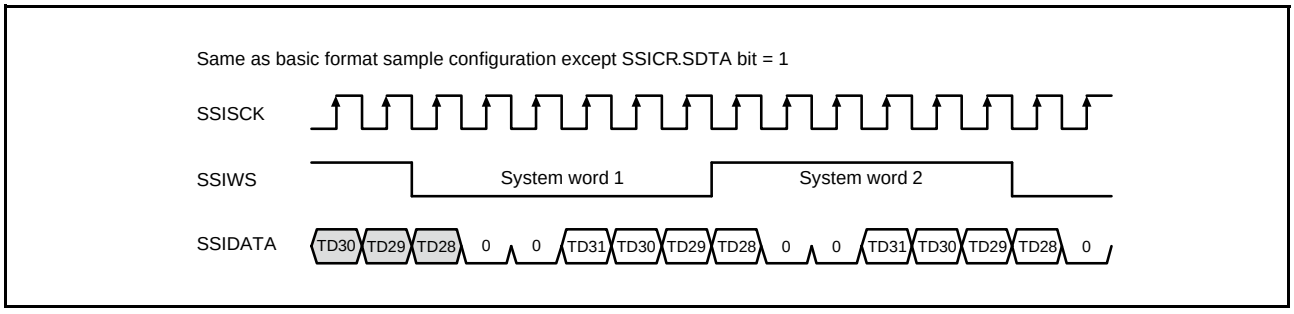



**Figure 37.10 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay**

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

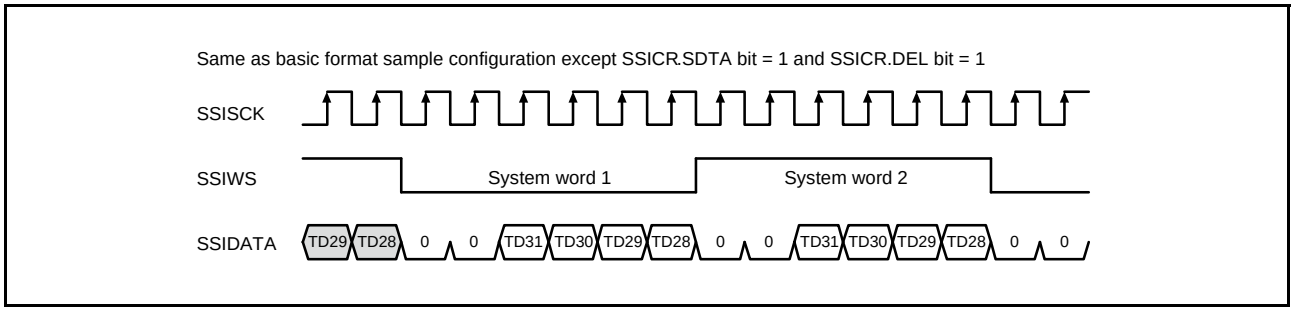



**Figure 37.11 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay**

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

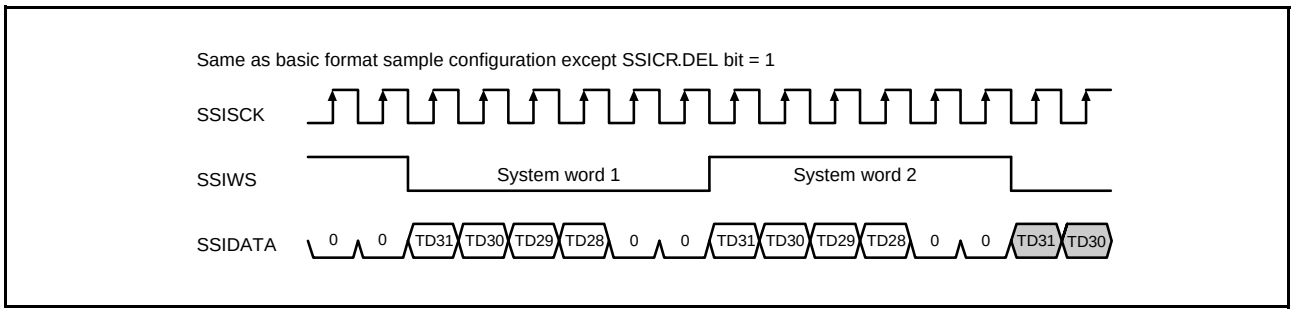



**Figure 37.12 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay**

- Parallel Right-Justified with Delay

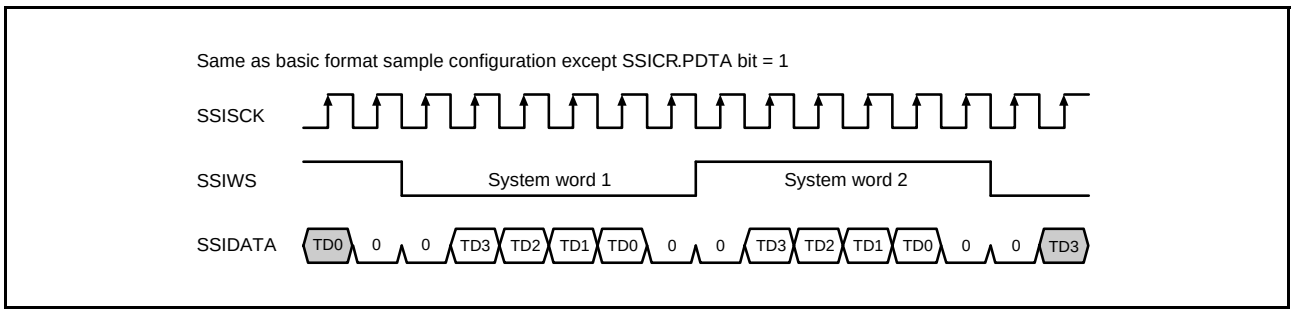



**Figure 37.13 Parallel Right-Justified with Delay**

- Mute Enabled

When the SSICR.MUEN bit is set to 1, the SSITXD0 pin becomes low (0) without synchronizing SSIWS0.

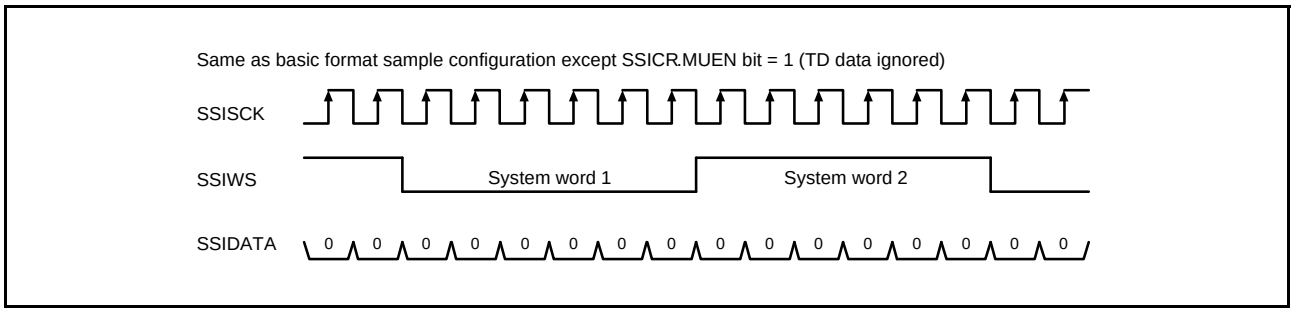



**Figure 37.14 Mute Enabled**

### 37.3.3 WS Continue Mode

In WS continue mode, the SSIWS0 signal continues to be toggled irrespective whether data transmission is enabled or disabled. This mode can be set using the SSITDMR.CONT bit. With this mode enabled, the SSIWS0 signal does not stop but continues toggling even if the SSICR.TEN and REN bits are both set to 0 (transmission disabled). With this mode disabled, the SSIWS0 signal stops if the SSICR.TEN and REN bits are both set to 0.

Figure 37.15 and Figure 37.16 show the operations with WS continue mode enabled and disabled, respectively.

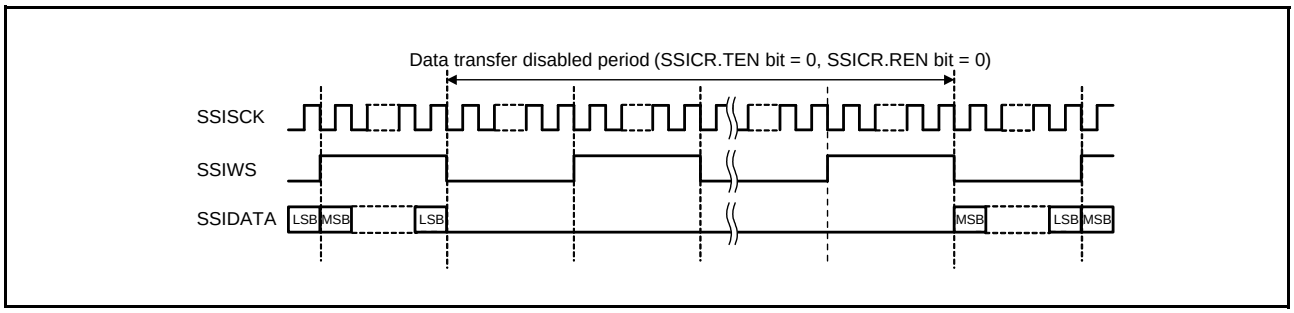



**Figure 37.15 WS Continue Mode Enabled**

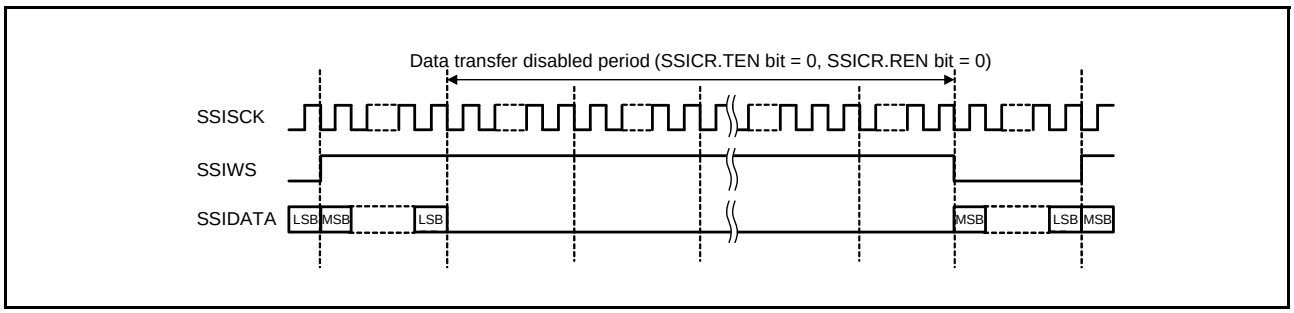



**Figure 37.16 WS Continue Mode Disabled**

### 37.3.4 Operating States

There are three states of operation: idle, communication, and waiting for idle. Figure 37.17 shows the operating state transitions.

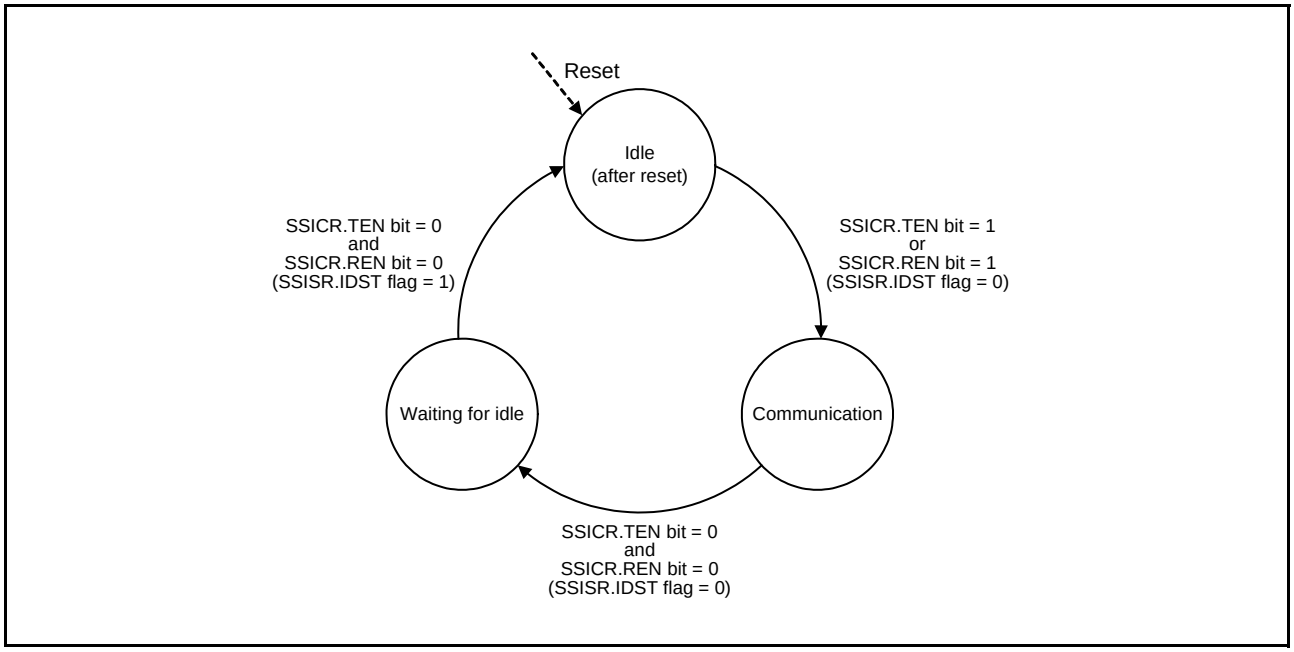



**Figure 37.17 Operating State Transitions**

#### (1) Idle State

This module enters this state when the MSTPCRD.MSTPD14 and MSTPD15 bits are set to 0 after a reset is released. All required configuration fields in the control register should be defined in this state. After the settings are made, the module enters communication state when the SSICR.TEN bit or SSICR.REN bit is set to 1.

#### (2) Communication State

Communication in this state depends on the selected operating state. For details, refer to section 37.3.5, Transmit Operation and section 37.3.6, Receive Operation.

#### (3) Waiting for Idle

This module enters this state when both the SSICR.TEN and SSICR.REN bits are set to 0 in communication state. If system word communication is completed in this state, the SSISR.IDST flag is set to 1 and this module enters the idle state.

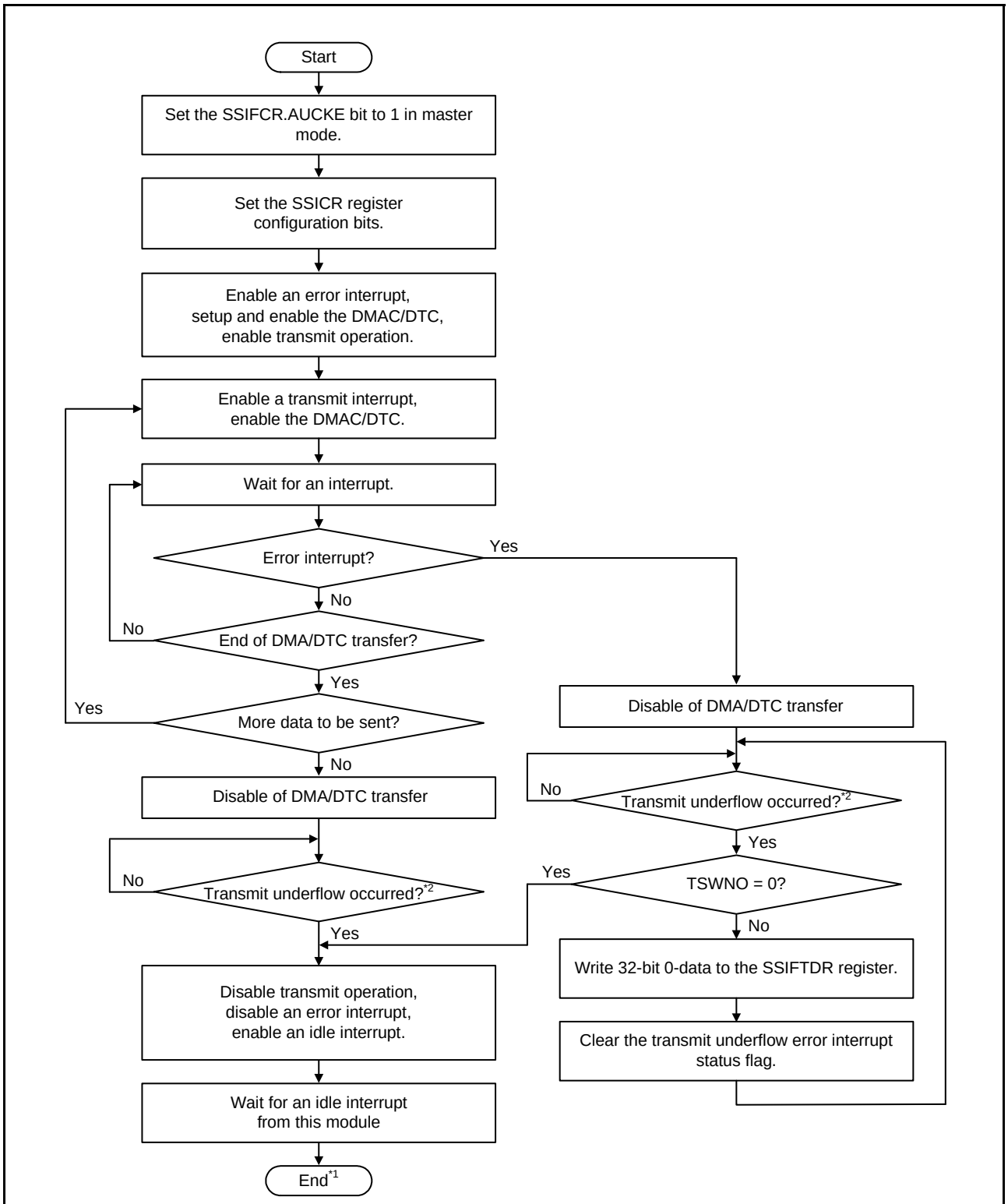
### 37.3.5 Transmit Operation

Transmission can be controlled either by DMA/DTC transfer or interrupt.
DMAC/DTC control is preferred to reduce the processor load. In transmission using the DMAC/DTC, the processor will only receive interrupts if there is an underflow or overflow of data or if DMA/DTC transfer has been completed. In transmission using DMA/DTC transfer, set the number of DMA/DTC transfers to multiples of 2 to write transmit data to the SSIFTDR register in 64-bit (two stages of FIFO) units.
The alternative method is using the interrupts that this module generates to supply data as required. In transmission using interrupts, write transmit data in 64-bit units regardless of the data format. If transmit data ends on a 32-bit boundary, write 00000000h after the last transmit data is written, and complete writing on a 64-bit boundary.
When stopping transmission, stop writing to the SSIFTDR register while 64-bit writing is completed. After writing is stopped, wait until a transmit underflow occurs before setting the SSICR.TEN bit to 0. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until this module enters the idle state. After setting the TEN bit to 0, continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state). If a transmit underflow error or transmit overflow error occurs during data transmission, transmit data to SSIFTDR register may not be written in a 64-bit units. In that case, stop writing data, wait until a transmit underflow error occurs, and check the SSISR.TSWNO flag when the transmit underflow has occurred. When the TSWNO flag is 1, write 00000000h to SSIFTDR register and wait until an underflow occurs again. Once the TSWNO flag is confirmed to be 0, Set the TEN bit to 0 and continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state).

Figure 37.18 shows transmission flow using the DMA/DTC, and Figure 37.19 shows transmission flow using interrupts.

Note 1. Input clock from the SSISCK0 pin when SSICR.SCKD bit = 0.
Master clock when SSICR.SCKD bit = 1.

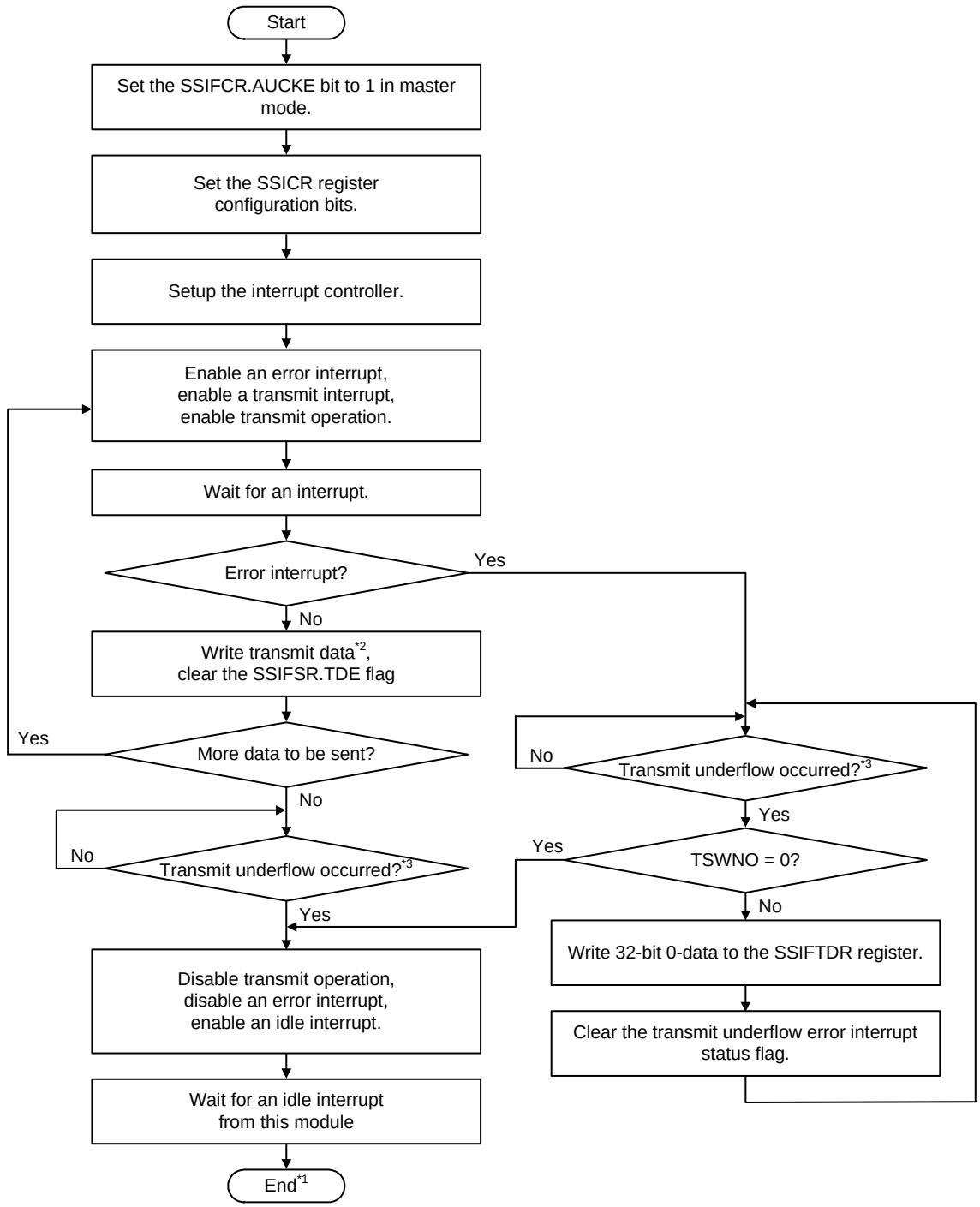
(1) Transmission Using the DMAC/DTC




Note 1. When restarting transmission after transmit operation is disabled (TEN = 0) while WS continue mode is disabled, execute a software reset and go back to the start of the flowchart again.
Note 2. Stop writing transmit data to the SSIFTDR register and wait until a transmit underflow occurs. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until the module is in the idle state.

**Figure 37.18 Transmission Using the DMAC/DTC**

## (2) Transmission Using Interrupts




Note 1. When restarting transmission after transmit operation is disabled (TEN = 0) while WS continue mode is disabled, execute a software reset and go back to the start of the flowchart again.
Note 2. Write transmit data in 64-bit units regardless of the data word setting.
Note 3. Stop writing transmit data to the SSIFTDR register and wait until a transmit underflow occurs. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until the module is in the idle state.

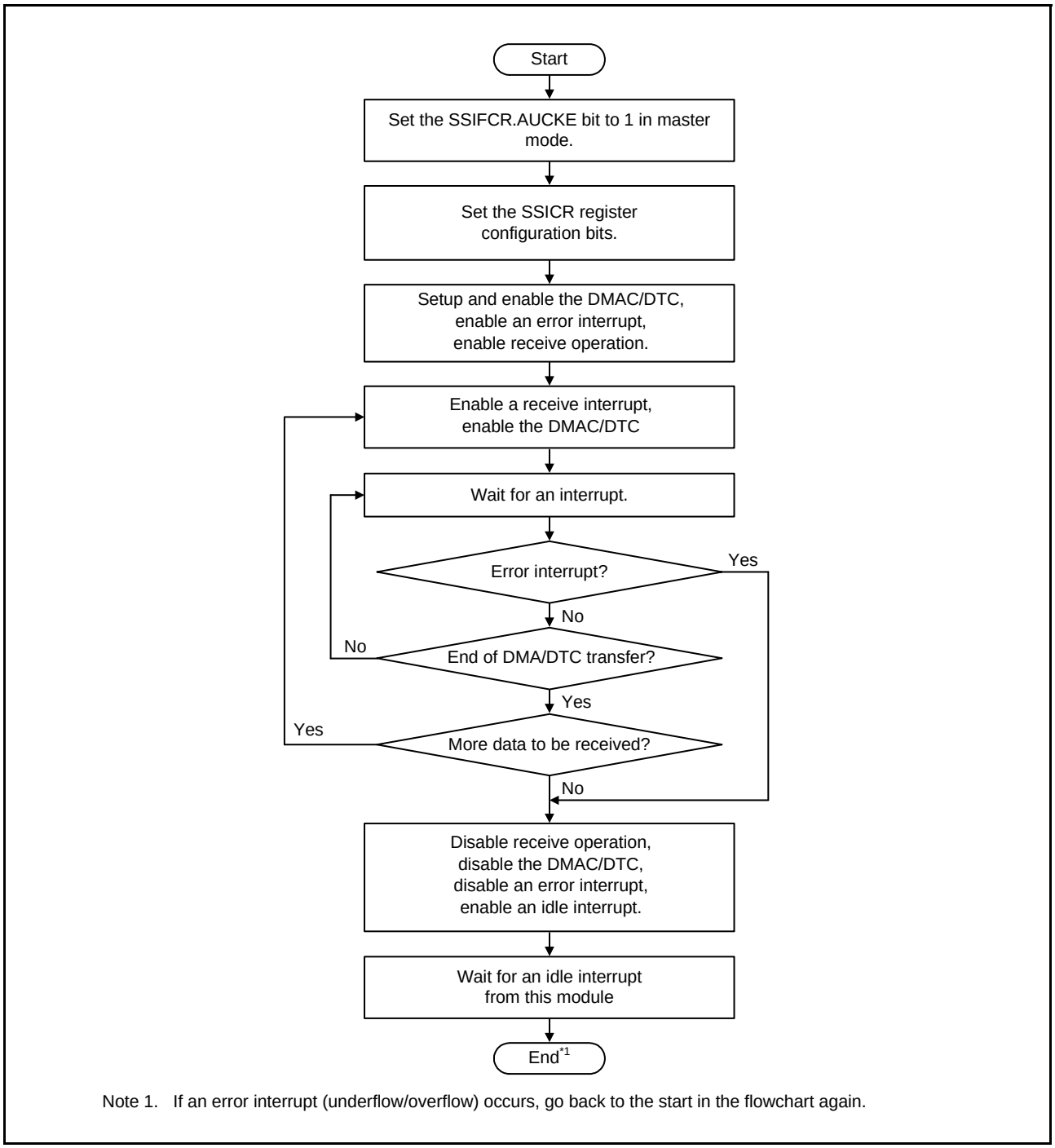
**Figure 37.19 Transmission Using Interrupts**

### 37.3.6 Receive Operation

Like transmission, reception can be controlled either by DMA/DTC transfer or interrupt.
Figure 37.20 and Figure 37.21 show the flow of operation.
When stopping reception, set the SSICR.REN bit to 0 and continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state).

Note 1. Input clock from the SSISCK0 pin when SSICR.SCKD bit = 0.
Master clock when SSICR.SCKD bit = 1.

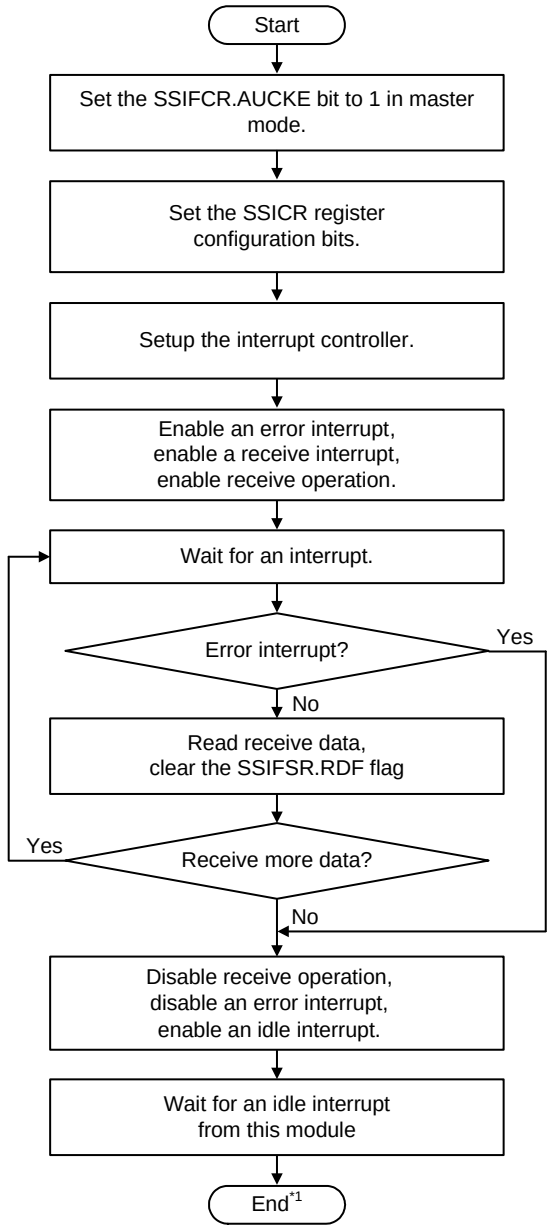
#### (1) Reception Using the DMAC/DTC




Note 1. If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.

**Figure 37.20 Reception Using the DMAC/DTC**

(2) Reception Using Interrupts




Note 1. If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.

**Figure 37.21 Reception Using Interrupts**

### 37.3.7 Serial Bit Clock Control

The SSI controls and selects the serial bit clock, according to the SSICR.SCKD and CKDV[3:0] bits setting.
If the serial bit clock direction is set to input (SCKD bit = 0), this module is in slave mode and the shift register uses the clock that was input to the SSISCK0 pin as the bit clock.

If the serial bit clock direction is set to output (SCKD bit = 1), this module is in master mode, and the shift register uses the master clock (MCLK) or a divided master clock as the bit clock. The master clock is divided by the ratio specified by the SSICR.CKDV[3:0] bits for use as the bit clock by the shift register.
In either case the module pin, SSISCK0, is the same as the bit clock.

## 37.4 Interrupt Sources

Table 37.7 lists the interrupt sources of the SSI. Each interrupt source can be enabled or disabled by the SSICR.TUIEN, TOIEN, RUIEN, ROIEN and IIEN bits, and the SSIFCR.TIE and RIE bits.

**Table 37.7 SSI Interrupt Sources**

| Channel | Interrupt Source | Description | Interrupt Status Flag | Interrupt Enable Bit | DMAC/DTC Start Trigger |
|---|---|---|---|---|---|
| SSI0 | SSIF0 | Transmit underflow interrupt<br>Transmit overflow interrupt<br>Receive underflow interrupt<br>Receive overflow interrupt<br>Idle interrupt | SSISR.TUIRQ<br>SSISR.TOIRQ<br>SSISR.RUIRQ<br>SSISR.ROIRQ<br>SSISR.IIRQ | SSICR.TUIEN<br>SSICR.TOIEN<br>SSICR.RUIEN<br>SSICR.ROIEN<br>SSICR.IIEN | Not available |
| | SSIRXI0 | Receive data full interrupt (RXI) | SSIFSR.RDF | SSIFCR.RIE | Available |
| | SSITXI0 | Transmit data empty interrupt (TXI) | SSIFSR.TDE | SSIFCR.TIE | Available |

## 37.5 Usage Notes

### 37.5.1 Setting the Module Stop Function

Module stop state can be entered or released using the MSTPCRD register. The initial setting of the SSI is in the module stop state. SSI register access is enabled by releasing the module stop state.
For details on the MSTPCRD register, refer to section 11, Low Power Consumption.

### 37.5.2 Notes on Changing Transmission Modes

For mode transitions between the transmitter, receiver, and transceiver while WS continue mode is disabled (SSITDMR.CONT = 0), set the SSICR.TEN and SSICR.REN bits to 0 and make a transition to the idle state once. Set the SSICR.TEN and SSICR.REN bits again while the module is in the idle state and restart transmission.

### 37.5.3 Limits on WS Continue Mode

If WS continue mode setting is changed, the operation of the SSISCK0 and SSIWS0 signals immediately after switching are not guaranteed. If it affects the device to be connected, do not change the setting dynamically.

# 38. Serial Peripheral Interface (RSPIa)

In this section, “PCLK” is used to refer to PCLKB.

## 38.1 Overview

This MCU includes one channel of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices.
Table 38.1 lists the specifications of the RSPI, and Figure 38.1 shows a block diagram of the RSPI.
In this section, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

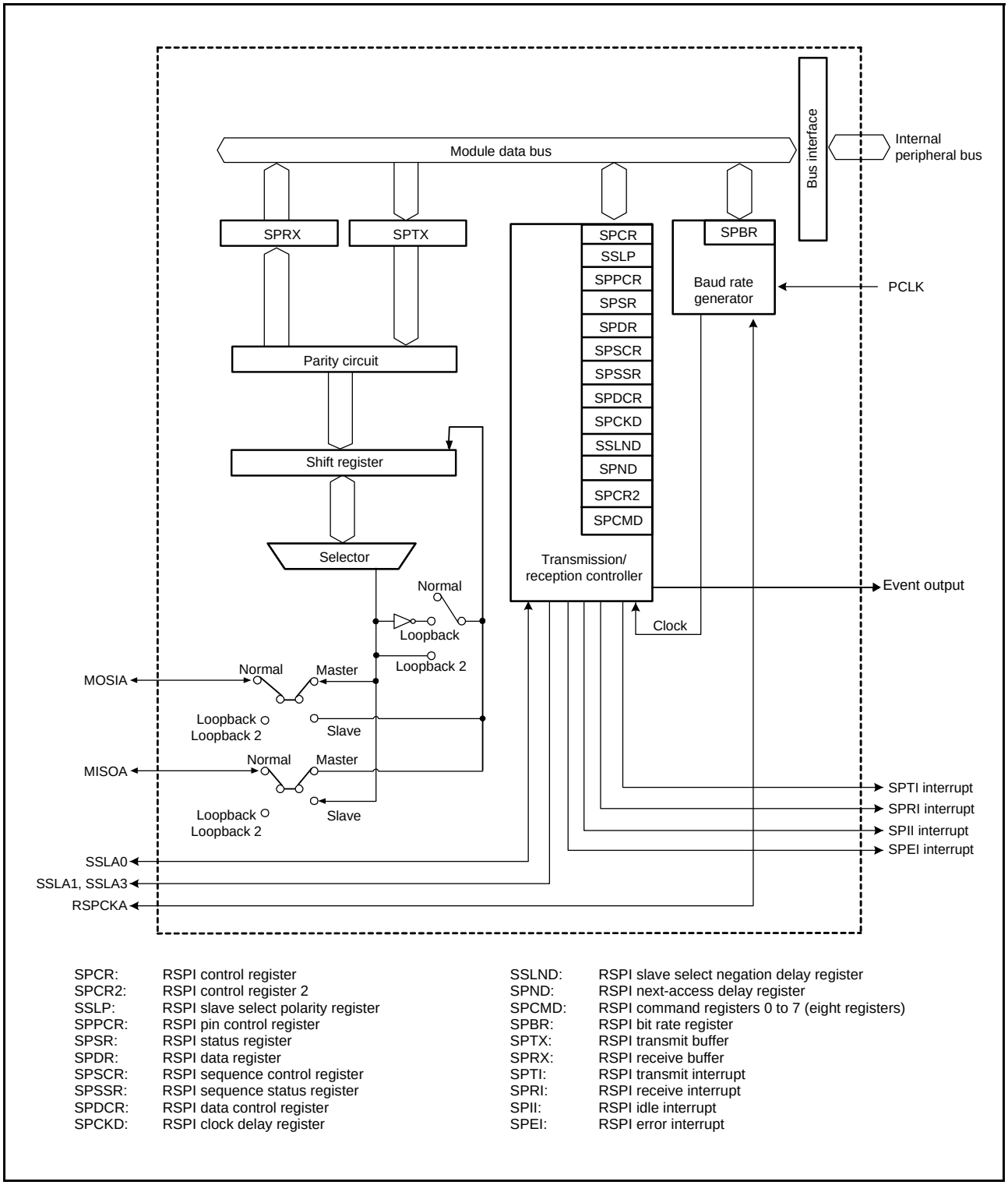
**Table 38.1 RSPI Specifications (1/2)**

| Item | Description |
|---|---|
| Number of channels | One channel |
| RSPI transfer functions | • Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method).<br>• Transmit-only operation is available.<br>• Communication mode: Full-duplex or transmit-only can be selected.<br>• Switching of the polarity of RSPCK<br>• Switching of the phase of RSPCK |
| Data format | • MSB first/LSB first selectable<br>• Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits.<br>• 128-bit transmit/receive buffers<br>• Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits). |
| Bit rate | • In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the division ratio ranges from divided by 2 to divided by 4096).<br>• In slave mode, the minimum PCLK clock divided by 8 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 8).<br>Width at high level: 4 cycles of PCLK; width at low level: 4 cycles of PCLK |
| Buffer configuration | • Double buffer configuration for the transmit/receive buffers<br>• 128 bits for the transmit/receive buffers |
| Error detection | • Mode fault error detection<br>• Overrun error detection*1<br>• Parity error detection |
| SSL control function | • Three SSL pins (SSLA0, SSLA1, and SSLA3) for each channel<br>• In single-master mode, SSLA0, SSLA1, and SSLA3 pins are output.<br>• In multi-master mode:<br>SSLA0 pin for input, and SSLA1, SSLA3 pins for either output or unused.<br>• In slave mode:<br>SSLA0 pin for input, and SSLA1, SSLA3 pins for unused.<br>• Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay)<br>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)<br>• Controllable delay from RSPCK stop to SSL output negation (SSL negation delay)<br>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)<br>• Controllable wait for next-access SSL output assertion (next-access delay)<br>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)<br>• Function for changing SSL polarity |
| Control in master transfer | • A transfer of up to eight commands can be executed sequentially in looped execution.<br>• For each command, the following can be set:<br>SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay<br>• A transfer can be initiated by writing to the transmit buffer.<br>• MOSI signal value specifiable in SSL negation<br>• RSPCK auto-stop function |

**Table 38.1 RSPI Specifications (2/2)**

| Item | Description |
|---|---|
| Interrupt sources | • Interrupt sources<br>Receive buffer full interrupt<br>Transmit buffer empty interrupt<br>RSPI error interrupt (mode fault, overrun, or parity error)<br>RSPI idle interrupt (RSPI idle) |
| Event link function (output) | • The following events can be output to the event link controller. (RSPI0)<br>Receive buffer full signal<br>Transmit buffer empty signal<br>Mode fault, overrun, or parity error signal<br>RSPI idle signal<br>Transmission-completed signal |
| Others | • Function for switching between CMOS output and open-drain output<br>• Function for initializing the RSPI<br>• Loopback mode |
| Low power consumption function | Module stop state can be set. |

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped at the timing of overrun error detection.




| | | | |
|---|---|---|---|
| SPCR: | RSPI control register | SSLND: | RSPI slave select negation delay register |
| SPCR2: | RSPI control register 2 | SPND: | RSPI next-access delay register |
| SSLP: | RSPI slave select polarity register | SPCMD: | RSPI command registers 0 to 7 (eight registers) |
| SPPCR: | RSPI pin control register | SPBR: | RSPI bit rate register |
| SPSR: | RSPI status register | SPTX: | RSPI transmit buffer |
| SPDR: | RSPI data register | SPRX: | RSPI receive buffer |
| SPSCR: | RSPI sequence control register | SPTI: | RSPI transmit interrupt |
| SPSSR: | RSPI sequence status register | SPRI: | RSPI receive interrupt |
| SPDCR: | RSPI data control register | SPII: | RSPI idle interrupt |
| SPCKD: | RSPI clock delay register | SPEI: | RSPI error interrupt |

**Figure 38.1 RSPI Block Diagram**

Table 38.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLA0 pin. SSLA0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin. Refer to section 38.3.2, Controlling RSPI Pins for details.

**Table 38.2 RSPI Pin Configuration**

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| RSPI0 | RSPCKA | I/O | Clock I/O |
| | MOSIA | I/O | Master transmit data I/O |
| | MISOA | I/O | Slave transmit data I/O |
| | SSLA0 | I/O | Slave selection I/O |
| | SSLA1 | Output | Slave selection output |
| | SSLA3 | Output | Slave selection output |

## 38.2 Register Descriptions

### 38.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SPRIE | SPE | SPTIE | SPEIE | MSTR | MODFEN | TXMD | SPMS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SPMS | RSPI Mode Select | 0: SPI operation (4-wire method)<br>1: Clock synchronous operation (3-wire method) | R/W |
| b1 | TXMD | Communications Operating Mode Select | 0: Full-duplex synchronous serial communications<br>1: Serial communications consisting of only transmit operations | R/W |
| b2 | MODFEN | Mode Fault Error Detection Enable | 0: Disables the detection of mode fault error<br>1: Enables the detection of mode fault error | R/W |
| b3 | MSTR | RSPI Master/Slave Mode Select | 0: Slave mode<br>1: Master mode | R/W |
| b4 | SPEIE | RSPI Error Interrupt Enable | 0: Disables the generation of RSPI error interrupt requests<br>1: Enables the generation of RSPI error interrupt requests | R/W |
| b5 | SPTIE | Transmit Buffer Empty Interrupt Enable | 0: Disables the generation of transmit buffer empty interrupt requests<br>1: Enables the generation of transmit buffer empty interrupt requests | R/W |
| b6 | SPE | RSPI Function Enable | 0: Disables the RSPI function<br>1: Enables the RSPI function | R/W |
| b7 | SPRIE | RSPI Receive Buffer Full Interrupt Enable | 0: Disables the generation of RSPI receive buffer full interrupt requests<br>1: Enables the generation of RSPI receive buffer full interrupt requests | R/W |

Do not change the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit while the SPCR.SPE bit is 1.

**SPMS Bit (RSPI Mode Select)**

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).
The SSLA0, SSLA1, and SSLA3 pins are not used in clock synchronous operation. The RSPCKA, MOSIA, and MISOA pins handle communications. If clock synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Do not set the CPHA bit to 0 when clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

**TXMD Bit (Communications Operating Mode Select)**

The TXMD bit selects full-duplex synchronous serial communications or transmit operations only.
When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (refer to section 38.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

#### MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 38.3.8, Error Detection). In addition, the RSPI determines the I/O direction of the SSLA0, SSLA1, and SSLA3 pins based on combinations of the MODFEN and MSTR bits (refer to section 38.3.2, Controlling RSPI Pins).

#### MSTR Bit (RSPI Master/Slave Mode Select)

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKA, MOSIA, MISOA, SSLA0, SSLA1, and SSLA3.

#### SPEIE Bit (RSPI Error Interrupt Enable)

The SPEIE bit enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the SPSR.MODF flag to 1, when the RSPI detects an overrun error and sets the SPSR.OVRF flag to 1, or when the RSPI detects a parity error and sets the SPSR.PERF flag to 1 (refer to section 38.3.8, Error Detection).

#### SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the RSPI detects when the transmit buffer is empty.
A transmit buffer empty interrupt request when transmission starts is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1.
Note that a transmit buffer interrupt is generated when the SPTIE bit is 1 even if the RSPI function is disabled (the SPTIE bit is changed to 0).

#### SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function.
When the SPSR.MODF flag is 1, the SPE bit cannot be set to 1. For details, refer to section 38.3.8, Error Detection.
Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 38.3.9, Initializing RSPI. Furthermore, a transmit buffer empty interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

#### SPRIE Bit (RSPI Receive Buffer Full Interrupt Enable)

If the RSPI has detected a receive buffer full write after completion of a serial transfer, the SPRIE bit enables or disables the generation of an RSPI receive buffer full interrupt request.

## 38.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 0008 8381h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | SSL3P | — | SSL1P | SSL0P |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SSL0P | SSL0 Signal Polarity Setting | 0: SSL0 signal is active low<br>1: SSL0 signal is active high | R/W |
| b1 | SSL1P | SSL1 Signal Polarity Setting | 0: SSL1 signal is active low<br>1: SSL1 signal is active high | R/W |
| b2 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b3 | SSL3P | SSL3 Signal Polarity Setting | 0: SSL3 signal is active low<br>1: SSL3 signal is active high | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Do not change the SSLP register while the SPCR.SPE bit is 1.

### 38.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 0008 8382h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | MOIFE | MOIFV | — | — | SPLP2 | SPLP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SPLP | RSPI Loopback | 0: Normal mode<br>1: Loopback mode (data is inverted for transmission) | R/W |
| b1 | SPLP2 | RSPI Loopback 2 | 0: Normal mode<br>1: Loopback mode (data is not inverted for transmission) | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | MOIFV | MOSI Idle Fixed Value | 0: The level output on the MOSIA pin during MOSI idling corresponds to low<br>1: The level output on the MOSIA pin during MOSI idling corresponds to high | R/W |
| b5 | MOIFE | MOSI Idle Value Fixing Enable | 0: MOSI output value equals final data from previous transfer<br>1: MOSI output value equals the value set in the MOIFV bit | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Do not change the SPPCR register while the SPCR.SPE bit is 1.

**SPLP Bit (RSPI Loopback)**

The SPLP bit selects the mode of the RSPI pins.

When the SPLP bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

**SPLP2 Bit (RSPI Loopback 2)**

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

**MOIFV Bit (MOSI Idle Fixed Value)**

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

**MOIFE Bit (MOSI Idle Value Fixing Enable)**

The MOIFE bit fixes the MOSIA output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.

## 38.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SPRF | — | SPTEF | — | PERF | MODF | IDLNF | OVRF |
| Value after reset: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | OVRF | Overrun Error Flag | 0: No overrun error occurs<br>1: An overrun error occurs | R/(W) *1 |
| b1 | IDLNF | RSPI Idle Flag | 0: RSPI is in the idle state<br>1: RSPI is in the transfer state | R |
| b2 | MODF | Mode Fault Error Flag | 0: No mode fault error occurs<br>1: A mode fault error occurs | R/(W) *1 |
| b3 | PERF | Parity Error Flag | 0: No parity error occurs<br>1: A parity error occurs | R/(W) *1 |
| b4 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5 | SPTEF | Transmit Buffer Empty Flag | 0: Transmit buffer has valid data<br>1: Transmit buffer has no valid data | R*2 |
| b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b7 | SPRF | Receive Buffer Full Flag | 0: Receive buffer has no valid data<br>1: Receive buffer has valid data | R*2 |

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. The write value should be 1.

### OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK clock auto-stop function is enabled (the SPCR2.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 38.3.8.1, Overrun Error.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then 0 is written to the OVRF flag.

### IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- Condition 1 and condition 2 are not satisfied in master mode under the [Clearing condition] below.

Slave mode

- The SPCR.SPE bit is 1 (enables the RSPI function)

[Clearing condition]

Master mode

- The following 1 is satisfied (condition 1) or all of the following 2 to 4 are satisfied (condition 2).

1. The SPCR.SPE bit is 0 (disables the RSPI function)
2. The transmit buffer (SPTX) is empty (data for the next transfer is not set)
3. The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)

4. The RSPI internal sequencer has entered the idle state (status in which operations up to the next-access delay have finished)

Slave mode

- The SPCR.SPE bit is 0 (disables the RSPI function)

#### MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error.

[Setting condition]

Multi-master mode

- When the input level of the SSLAi pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

Slave mode

- When the SSLAi pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

The active level of the SSLAi signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then 0 is written to the MODF flag

#### PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error

[Clearing condition]

- When SPSR is read while the PERF flag is 1, and then 0 is written to the PERF flag

#### SPTEF Flag (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer (SPTX) in the RSPI data register has valid data.

[Setting condition]

- When the SPCR.SPE bit is 0 (disables the RSPI function)
- When data is transferred from the transmit buffer to the shift register

[Clearing condition]

- When the number of frames of transmit data specified by the SPDCR.SPFC[1:0] bits is written to the SPDR register

The SPDR register can be set only when the SPTEF flag is 1. The data in the transmit buffer is not updated when the SPDR register is set while the SPTEF flag is 0.

#### SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.TXMD bit is 0 (full duplex) and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.

[Clearing condition]

- When all of the received data are read from the SPDR register

## 38.2.5 RSPI Data Register (SPDR)

- When accessing in longword size

Address(es): RSPI0.SPDR 0008 8384h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

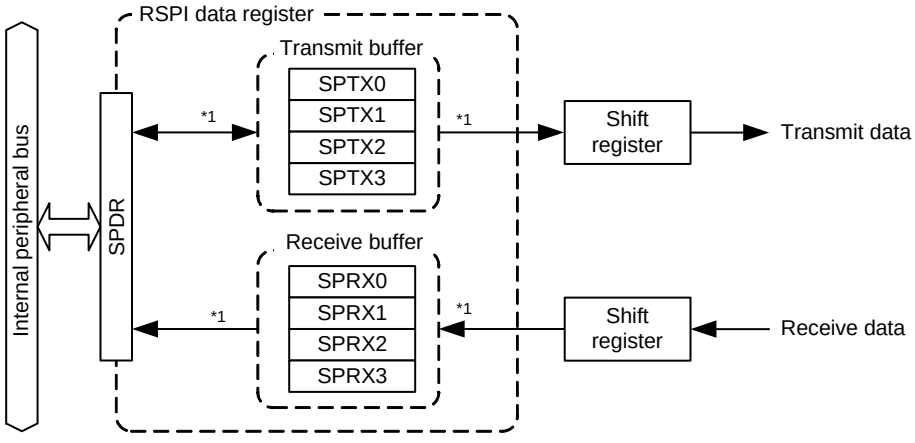
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- When accessing in word size

Address(es): RSPI0.SPDR.H 0008 8384h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SPDR is the interface with the buffers that hold data for transmission and reception by the RSPI.
When accessing in longwords (the SPLW bit is 1), access SPDR in 32-bit units.
When accessing in words (the SPLW bit is 0), access SPDR.H in 16-bit units.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 38.2 shows the Configuration of SPDR.




Note 1. The destination buffer and stage for access is automatically switched by hardware.

**Figure 38.2 Configuration of SPDR**

The transmit and receive buffers each have four stages. The number of stages to be used is selectable by the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all

mapped to the single address of SPDR.
Data written to SPDR are written to a transmit-buffer stage (SPTXn) (n = 0 to 3) and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.
Furthermore, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the corresponding bits in SPRXn. For example, if the data length is 9 bits, received data are stored in the SPRXn[8:0] bits and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

### (1) Bus Interface

SPDR is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for SPDR. Furthermore, the unit of access for SPDR is selected by the SPDCR.SPLW bit.
Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end. Operations involved in writing to and reading from SPDR are described below.

#### (a) Writing

Data written to SPDR are written to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from SPDR.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to SPDR.
Figure 38.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.

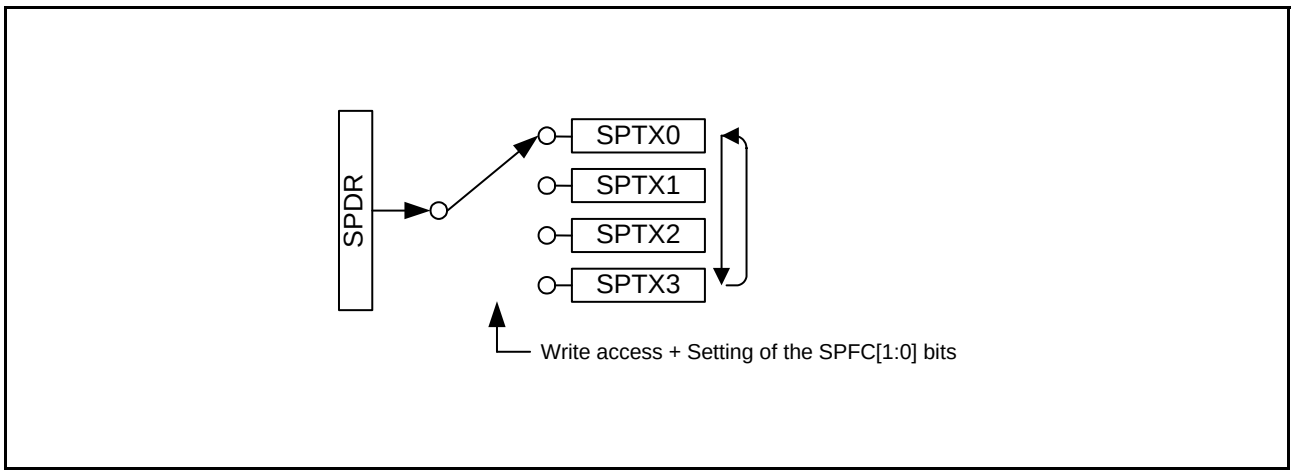



**Figure 38.3 Configuration of SPDR (Writing)**

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer among SPTX0 to SPTX3.
  When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → …
  When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPTX0 will be the destination the next time writing proceeds.
When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames set by the number of frames specification bits (SPFC[1:0]) in

the RSPI data control register (SPDCR). Even if the number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (while the SPSR.SPTEF flag is 0).

### (b) Reading

SPDR can be read to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.

Figure 38.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

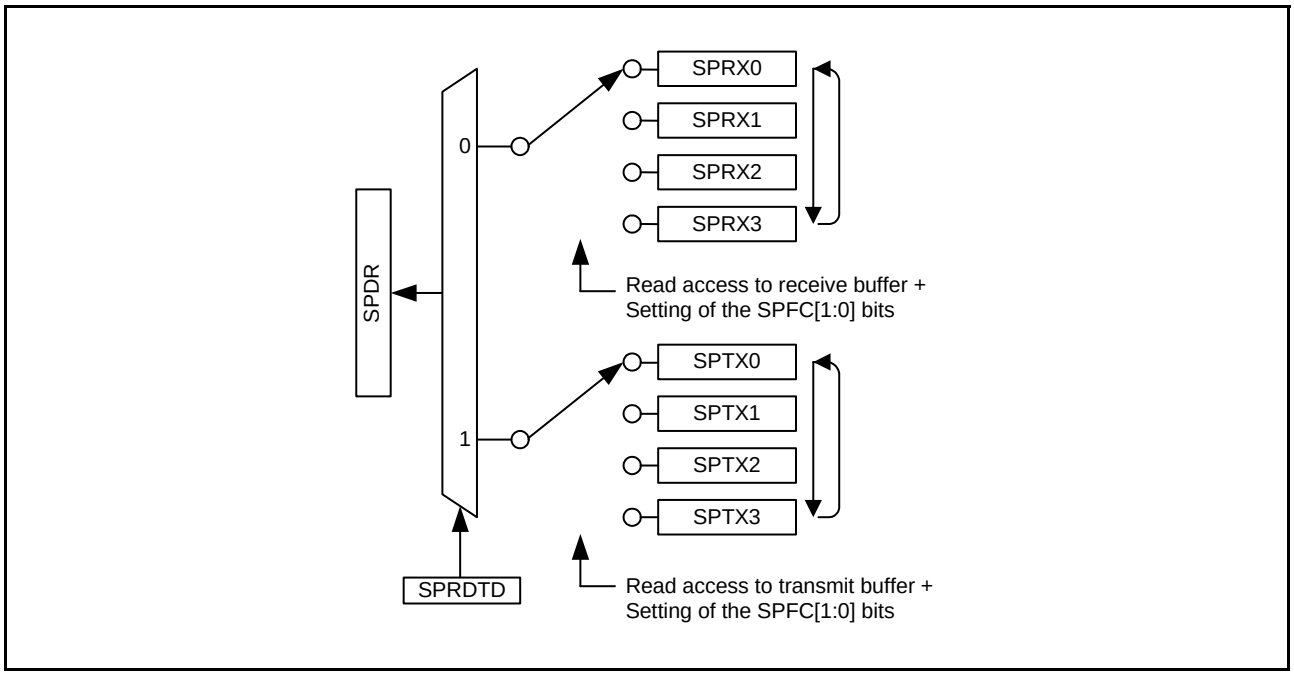



**Figure 38.4 Configuration of SPDR (Reading)**

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.

The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

The transmit buffer read pointer is updated when writing to SPDR, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0 in the interval after completion of writing the number of frames of data specified in the number of frames specification bits (SPDCR.SPFC[1:0]) and before generation of the next buffer empty interrupt (while the SPSR.SPTEF flag is 0).

## 38.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 0008 8388h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SPSLN[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SPSLN[2:0] | RSPI Sequence Length Specification | b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 (No.)<br>0 0 0: 1 — 0→0→…<br>0 0 1: 2 — 0→1→0→…<br>0 1 0: 3 — 0→1→2→0→…<br>0 1 1: 4 — 0→1→2→3→0→…<br>1 0 0: 5 — 0→1→2→3→4→0→…<br>1 0 1: 6 — 0→1→2→3→4→5→0→…<br>1 1 0: 7 — 0→1→2→3→4→5→6→0→…<br>1 1 1: 8 — 0→1→2→3→4→5→6→7→0→…<br>The order in which the SPCMD0 to SPCMD7 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPI is shown above. However, the RSPI in slave mode references SPCMD0. | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

SPSCR sets the sequence length when the RSPI operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

**SPSLN[2:0] Bits (RSPI Sequence Length Specification)**

The SPSLN[2:0] bits specify a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits.
In slave mode, SPCMD0 is referred.

## 38.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 0008 8389h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | SPECM[2:0] | | | — | SPCP[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SPCP[2:0] | RSPI Command Pointer | b2 b0<br>0 0 0: SPCMD0<br>0 0 1: SPCMD1<br>0 1 0: SPCMD2<br>0 1 1: SPCMD3<br>1 0 0: SPCMD4<br>1 0 1: SPCMD5<br>1 1 0: SPCMD6<br>1 1 1: SPCMD7 | R |
| b3 | — | Reserved | This bit is read as 0. | R |
| b6 to b4 | SPECM[2:0] | RSPI Error Command | b6 b4<br>0 0 0: SPCMD0<br>0 0 1: SPCMD1<br>0 1 0: SPCMD2<br>0 1 1: SPCMD3<br>1 0 0: SPCMD4<br>1 0 1: SPCMD5<br>1 1 0: SPCMD6<br>1 1 1: SPCMD7 | R |
| b7 | — | Reserved | This bit is read as 0. | R |

SPSSR indicates the sequence control status when the RSPI operates in master mode.
Any writing to SPSSR is ignored.

### SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate SPCMDm that is currently pointed to by the pointer during sequence control by the RSPI. For the RSPI's sequence control, refer to section 38.3.10.1, Master Mode Operation.

### SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate SPCMDm that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPI. The RSPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning. For the RSPI's error detection function, refer to section 38.3.8, Error Detection. For the RSPI's sequence control, refer to section 38.3.10.1, Master Mode Operation.

## 38.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SPBR sets the bit rate in master mode. Do not change the SPBR register while both the SPCR.MSTR and SPCR.SPE bits are 1.

When the RSPI is used in slave mode, the bit rate depends on the bit rate of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).
The bit rate is determined by combinations of the SPBR setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(PCLK)}{2 \times (n+1) \times 2^{N}}$$

Table 38.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.
Use the bit rate that meets electrical characteristics based on the AC specifications of the target device.

**Table 38.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates**

| SPBR (n) | BRDV[1:0] Bits (N) | Division Ratio | Bit Rate |
|---|---|---|---|
| | | | PCLK = 32 MHz |
| 0 | 0 | 2 | 16.0 Mbps |
| 1 | 0 | 4 | 8.00 Mbps |
| 2 | 0 | 6 | 5.33 Mbps |
| 3 | 0 | 8 | 4.00 Mbps |
| 4 | 0 | 10 | 3.20 Mbps |
| 5 | 0 | 12 | 2.67 Mbps |
| 5 | 1 | 24 | 1.33 Mbps |
| 5 | 2 | 48 | 667 kbps |
| 5 | 3 | 96 | 333 kbps |
| 255 | 3 | 4096 | 7.81 kbps |

## 38.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | SPLW | SPRDTD | — | — | SPFC[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | SPFC[1:0] | Number of Frames Specification | b1 b0<br>0 0: 1 frame<br>0 1: 2 frames<br>1 0: 3 frames<br>1 1: 4 frames | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | SPRDTD | RSPI Receive/Transmit Data Select | 0: SPDR values are read from the receive buffer<br>1: SPDR values are read from the transmit buffer (but only if the transmit buffer is empty) | R/W |
| b5 | SPLW | RSPI Longword Access/ Word Access Specification | 0: SPDR is accessed in words<br>1: SPDR is accessed in longwords | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Up to four frames can be transmitted or received in one round of transmission or reception activation. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.
When changing the SPDCR.SPFC[1:0] bits while the SPCR.SPE bit is 1, the bits should be changed while the SPSR.IDLNF flag is 0.

### SPFC[1:0] Bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR (per transfer activation). Up to four frames can be transmitted or received in one round of transmission or reception, and the amount of data is determined by the combination of the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits. Furthermore, the setting of the SPFC[1:0] bits adjusts the number of frames for generation of RSPI receive buffer full interrupt, and start of transmission or generation of transmit buffer empty interrupts.
When the number of frames of transmit data specified by SPFC[1:0] bits is written to the SPDR register, the SPSR.SPTEF flag becomes 0 and transmission starts. Then, when the specified number of frames of transmit data has been transferred to the shift register, the SPTEF flag becomes 1 and the RSPI transmit buffer empty interrupt is generated.
When the number of frames specified by the SPFC[1:0] bits are received, the SPSR.SPRF flag becomes 1 and the RSPI receive buffer full interrupt is generated.
Table 38.4 lists the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. Do not select the combinations of settings other than those shown in the examples.

**Table 38.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits**

| Setting | SPSLN[2:0] | SPFC[1:0] | Number of Frames in a Single Sequence | Number of Frames at which Transmit Buffer or Receive Buffer Status Becomes “Has Valid Data” |
|---|---|---|---|---|
| 1-1 | 000b | 00b | 1 | 1 |
| 1-2 | 000b | 01b | 2 | 2 |
| 1-3 | 000b | 10b | 3 | 3 |
| 1-4 | 000b | 11b | 4 | 4 |
| 2-1 | 001b | 01b | 2 | 2 |
| 2-2 | 001b | 11b | 4 | 4 |
| 3 | 010b | 10b | 3 | 3 |
| 4 | 011b | 11b | 4 | 4 |
| 5 | 100b | 00b | 5 | 1 |
| 6 | 101b | 00b | 6 | 1 |
| 7 | 110b | 00b | 7 | 1 |
| 8 | 111b | 00b | 8 | 1 |

### SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR reads values from the receive buffer or from the transmit buffer.
If reading is from the transmit buffer, the value written to SPDR register immediately beforehand is read.
When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt (While the SPSR.SPTEF flag is 1).
For details, refer to section 38.2.5, RSPI Data Register (SPDR).

### SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for SPDR. Access to the SPDR register in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 to 16 bits. Do not select 20, 24, or 32 bits.

## 38.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 0008 838Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SCKDL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SCKDL[2:0] | RSPCK Delay Setting | b2 b0<br>0 0 0: 1 RSPCK<br>0 0 1: 2 RSPCK<br>0 1 0: 3 RSPCK<br>0 1 1: 4 RSPCK<br>1 0 0: 5 RSPCK<br>1 0 1: 6 RSPCK<br>1 1 0: 7 RSPCK<br>1 1 1: 8 RSPCK | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

SPCKD sets a period from the beginning of SSLAi signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. Do not change the SPCKD register while both the SPCR.MSTR and SPCR.SPE bits are 1.

### SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000b.

## 38.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 0008 838Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SLNDL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SLNDL[2:0] | SSL Negation Delay Setting | b2 b0<br>0 0 0: 1 RSPCK<br>0 0 1: 2 RSPCK<br>0 1 0: 3 RSPCK<br>0 1 1: 4 RSPCK<br>1 0 0: 5 RSPCK<br>1 0 1: 6 RSPCK<br>1 1 0: 7 RSPCK<br>1 1 1: 8 RSPCK | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLAi signal during a serial transfer by the RSPI in master mode. Do not change the SSLND register while both the SPCR.MSTR and SPCR.SPE bits are 1.

**SLNDL[2:0] Bits (SSL Negation Delay Setting)**

The SLNDL[2:0] bits set an SSL negation delay value when the RSPI is in master mode.
When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

## 38.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 0008 838Eh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SPNDL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SPNDL[2:0] | RSPI Next-Access Delay Setting | b2 b0<br>0 0 0: 1 RSPCK + 2 PCLK<br>0 0 1: 2 RSPCK + 2 PCLK<br>0 1 0: 3 RSPCK + 2 PCLK<br>0 1 1: 4 RSPCK + 2 PCLK<br>1 0 0: 5 RSPCK + 2 PCLK<br>1 0 1: 6 RSPCK + 2 PCLK<br>1 1 0: 7 RSPCK + 2 PCLK<br>1 1 1: 8 RSPCK + 2 PCLK | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

SPND sets a non-active period (next-access delay) of the SSLAi signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. Do not change the SPND register while both the SPCR.MSTR and SPCR.SPE bits are 1.

**SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)**

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000b.

## 38.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 0008 838Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | SCKASE | PTE | SPIIE | SPOE | SPPE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SPPE | Parity Enable | 0: Does not add the parity bit to transmit data and does not check the parity bit of receive data<br>1: Adds the parity bit to transmit data and checks the parity bit of receive data (when SPCR.TXMD = 0)<br>Adds the parity bit to transmit data but does not check the parity bit of receive data (when SPCR.TXMD = 1) | R/W |
| b1 | SPOE | Parity Mode | 0: Selects even parity for use in transmission and reception<br>1: Selects odd parity for use in transmission and reception | R/W |
| b2 | SPIIE | RSPI Idle Interrupt Enable | 0: Disables the generation of idle interrupt requests<br>1: Enables the generation of idle interrupt requests | R/W |
| b3 | PTE | Parity Self-Diagnosis | 0: Disables the self-diagnosis function of the parity circuit<br>1: Enables the self-diagnosis function of the parity circuit | R/W |
| b4 | SCKASE | RSPCK Auto-Stop Function Enable | 0: Disables the RSPCK auto-stop function<br>1: Enables the RSPCK auto-stop function | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Do not change the SPPE, SPOE, and SCKASE bits in the SPCR2 register while the SPCR.SPE bit is 1.

### SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.

The parity bit is added to transmit data and parity checking is performed for receive data when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1.

The parity bit is added to transmit data but parity checking is not performed for receive data when the SPCR.TXMD bit is 1 and the SPCR2.SPPE bit is 1.

### SPOE Bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd.

The SPOE bit is valid only when the SPPE bit is 1.

### SPIIE Bit (RSPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of RSPI idle interrupt requests when the RSPI being in the idle state is detected and the SPSR.IDLNF flag is set to 0.

### PTE Bit (Parity Self-Diagnosis)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

**SCKASE Bit (RSPCK Auto-Stop Function Enable)**

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs when data is received in master mode. For details, refer to section 38.3.8.1, Overrun Error.

## 38.2.14 RSPI Command Register m (SPCMDm) (m = 0 to 7)

Address(es): RSPI0.SPCMD0 0008 8390h, RSPI0.SPCMD1 0008 8392h, RSPI0.SPCMD2 0008 8394h, RSPI0.SPCMD3 0008 8396h, RSPI0.SPCMD4 0008 8398h, RSPI0.SPCMD5 0008 839Ah, RSPI0.SPCMD6 0008 839Ch, RSPI0.SPCMD7 0008 839Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SCKDEN | SLNDEN | SPNDEN | LSBF | SPB[3:0] | | | | SSLKP | SSLA[2:0] | | | BRDV[1:0] | | CPOL | CPHA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CPHA | RSPCK Phase Setting | 0: Data sampling on odd edge, data variation on even edge<br>1: Data variation on odd edge, data sampling on even edge | R/W |
| b1 | CPOL | RSPCK Polarity Setting | 0: RSPCK is low when idle<br>1: RSPCK is high when idle | R/W |
| b3, b2 | BRDV[1:0] | Bit Rate Division Setting | b3 b2<br>0 0: These bits select the base bit rate<br>0 1: These bits select the base bit rate divided by 2<br>1 0: These bits select the base bit rate divided by 4<br>1 1: These bits select the base bit rate divided by 8 | R/W |
| b6 to b4 | SSLA[2:0] | SSL Signal Assertion Setting | b6 b4<br>0 0 0: SSL0<br>0 0 1: SSL1<br>0 1 1: SSL3<br>1 x x: Setting prohibited | R/W |
| b7 | SSLKP | SSL Signal Level Keeping | 0: Negates all SSL signals upon completion of transfer<br>1: Keeps the SSL signal level from the end of transfer until the beginning of the next access | R/W |
| b11 to b8 | SPB[3:0] | RSPI Data Length Setting | b11 b8<br>0100 to 0111: 8 bits<br>1 0 0 0: 9 bits<br>1 0 0 1: 10 bits<br>1 0 1 0: 11 bits<br>1 0 1 1: 12 bits<br>1 1 0 0: 13 bits<br>1 1 0 1: 14 bits<br>1 1 1 0: 15 bits<br>1 1 1 1: 16 bits<br>0 0 0 0: 20 bits<br>0 0 0 1: 24 bits<br>0010, 0011: 32 bits | R/W |
| b12 | LSBF | RSPI LSB First | 0: MSB first<br>1: LSB first | R/W |
| b13 | SPNDEN | RSPI Next-Access Delay Enable | 0: A next-access delay of 1 RSPCK + 2 PCLK<br>1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND) | R/W |
| b14 | SLNDEN | SSL Negation Delay Setting Enable | 0: An SSL negation delay of 1 RSPCK<br>1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND) | R/W |
| b15 | SCKDEN | RSPCK Delay Setting Enable | 0: An RSPCK delay of 1 RSPCK<br>1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD) | R/W |

x: Don’t care

SPCMDm register is used to set a transfer format for the RSPI in master mode. Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). Some of the bits in SPCMD0 register is used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMDm register according to the settings in the SPSCR.SPSLN[2:0] bits, and executes the serial transfer that is set in the referenced SPCMDm register.
SPCMDm register should be set while the transmit buffer is empty (data for the next transfer is not set) and before setting of the data that is to be transmitted when that SPCMDm register is referenced.
SPCMDm that is referenced by the RSPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. Do not change the SPCMDm register while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1.

#### CPHA Bit (RSPCK Phase Setting)

The CPHA bit sets an RSPCK phase of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.

#### CPOL Bit (RSPCK Polarity Setting)

The CPOL bit sets an RSPCK polarity of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.

#### BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and SPBR (refer to section 38.2.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

#### SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLAi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLA0 pin acts as input).
When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

#### SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLAi signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 38.3.10.1, Master Mode Operation (4) Burst Transfer.
When using the RSPI in slave mode, the SSLKP bit should be set to 0.

#### SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode. When the SPDCR.SPLW bit is 0, set the SPB[3:0] bits to 0100b (8 bits) to 1111b (16 bits).

#### LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

**SPNDEN Bit (RSPI Next-Access Delay Enable)**

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSLAi signal inactive until the RSPI enables the SSLAi signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2 PCLK. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.

When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

**SLNDEN Bit (SSL Negation Delay Setting Enable)**

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLAi signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

**SCKDEN Bit (RSPCK Delay Setting Enable)**

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLAi signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

## 38.3 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

### 38.3.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave mode (SPI operation), single-master mode (SPI operation), multi-master mode (SPI operation), slave mode (clock synchronous operation), and master mode (clock synchronous operation). A particular mode of the RSPI can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR. Table 38.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

**Table 38.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode**

| | **SPI Operation** | | | **Clock Synchronous Operation** | |
|---|---|---|---|---|---|
| **Mode** | **Slave** | **Single-Master** | **Multi-Master** | **Slave** | **Master** |
| MSTR bit setting | 0 | 1 | 1 | 0 | 1 |
| MODFEN bit setting | 0 or 1 | 0 | 1 | 0 | 0 |
| SPMS bit setting | 0 | 0 | 0 | 1 | 1 |
| RSPCKA signal | Input | Output | Output/Hi-Z | Input | Output |
| MOSIA signal | Input | Output | Output/Hi-Z | Input | Output |
| MISOA signal | Output/Hi-Z | Input | Input | Output | Input |
| SSLA0 signal | Input | Output | Input | Hi-Z*1 | Hi-Z*1 |
| SSLA1, SSLA3 signals | Hi-Z*1 | Output | Output/Hi-Z | Hi-Z*1 | Hi-Z*1 |
| SSL polarity change function | Supported | Supported | Supported | — | — |
| Transfer rate | Up to PCLK/8 | Up to PCLK/2 | Up to PCLK/2 | Up to PCLK/8 | Up to PCLK/2 |
| Clock source | RSPCK input | On-chip baud rate generator | On-chip baud rate generator | RSPCK input | On-chip baud rate generator |
| Clock polarity | Two | | | | |
| Clock phase | Two | Two | Two | One (CPHA = 1) | Two |
| First transfer bit | MSB/LSB | | | | |
| Transfer data length | 8 to 16, 20, 24, 32 bits | | | | |
| Burst transfer | Possible (CPHA = 1) | Possible (CPHA = 0,1) | Possible (CPHA = 0,1) | — | — |
| RSPCK delay control | Not supported | Supported | Supported | Not supported | Supported |
| SSL negation delay control | Not supported | Supported | Supported | Not supported | Supported |
| Next-access delay control | Not supported | Supported | Supported | Not supported | Supported |
| Transfer activation method | SSL input active or RSPCK oscillation | Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1 | Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1 | RSPCK oscillation | Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1 |
| Sequence control | Not supported | Supported | Supported | Not supported | Supported |
| Transmit buffer empty detection | Supported | | | | |
| Receive buffer full detection | Supported*2 | | | | |
| Overrun error detection | Supported*2 | Supported*2, *4 | Supported*2, *4 | Supported*2 | Supported*2 |
| Parity error detection | Supported*2, *3 | | | | |
| Mode fault error detection | Supported (MODFEN = 1) | Not supported | Supported | Not supported | Not supported |

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.
Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR2.SCKASE bit is 1, overrun error detection does not proceed.

### 38.3.2 Controlling RSPI Pins

According to the MSTR, MODFEN, and SPMS bits in SPCR and the ODRn.Bi bit for I/O ports, the RSPI can switch pin states. Table 38.6 lists the relationship between pin states and bit settings. Setting the ODRn.Bi bit for an I/O port to 0 selects CMOS output; setting it to 1 selects open-drain output. The I/O port settings should follow this relationship.

**Table 38.6 Relationship between Pin States and Bit Settings**

| Mode | Pin | Pin State*2 | |
|---|---|---|---|
| | | ODRn.Bi Bit for I/O Ports = 0 | ODRn.Bi Bit for I/O Ports = 1 |
| Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0) | RSPCKA | CMOS output | Open-drain output |
| | SSLA0, SSLA1, SSLA3 | CMOS output | Open-drain output |
| | MOSIA | CMOS output | Open-drain output |
| | MISOA | Input | Input |
| Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0) | RSPCKA*3 | CMOS output/Hi-Z | Open-drain output/Hi-Z |
| | SSLA0 | Input | Input |
| | SSLA1, SSLA3*3 | CMOS output/Hi-Z | Open-drain output/Hi-Z |
| | MOSIA*3 | CMOS output/Hi-Z | Open-drain output/Hi-Z |
| | MISOA | Input | Input |
| Slave mode (SPI operation) (MSTR = 0, SPMS = 0) | RSPCKA | Input | Input |
| | SSLA0 | Input | Input |
| | SSLA1, SSLA3*5 | Hi-Z*1 | Hi-Z*1 |
| | MOSIA | Input | Input |
| | MISOA*4 | CMOS output/Hi-Z | Open-drain output/Hi-Z |
| Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1) | RSPCKA | CMOS output | Open-drain output |
| | SSLA0, SSLA1, SSLA3*5 | Hi-Z*1 | Hi-Z*1 |
| | MOSIA | CMOS output | Open-drain output |
| | MISOA | Input | Input |
| Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1) | RSPCKA | Input | Input |
| | SSLA0, SSLA1, SSLA3*5 | Hi-Z*1 | Hi-Z*1 |
| | MOSIA | Input | Input |
| | MISOA | CMOS output | Open-drain output |

Note 1. This function is not supported in this mode.
Note 2. RSPI settings are not reflected in the multiplex pins for which the RSPI function is not selected.
Note 3. When SSLA0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLA0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The RSPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as listed in Table 38.7.

**Table 38.7 MOSI Signal Value Determination during SSL Negation Period**

| MOIFE Bit | MOIFV Bit | MOSIA Signal Value during SSL Negation Period |
|---|---|---|
| 0 | 0, 1 | Final data from previous transfer |
| 1 | 0 | Low |
| 1 | 1 | High |

### 38.3.3 RSPI System Configuration Examples

#### 38.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 38.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLA0, SSLA1, and SSLA3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state.*1
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLAi output of this MCU should be connected to the SSL input of the slave device.

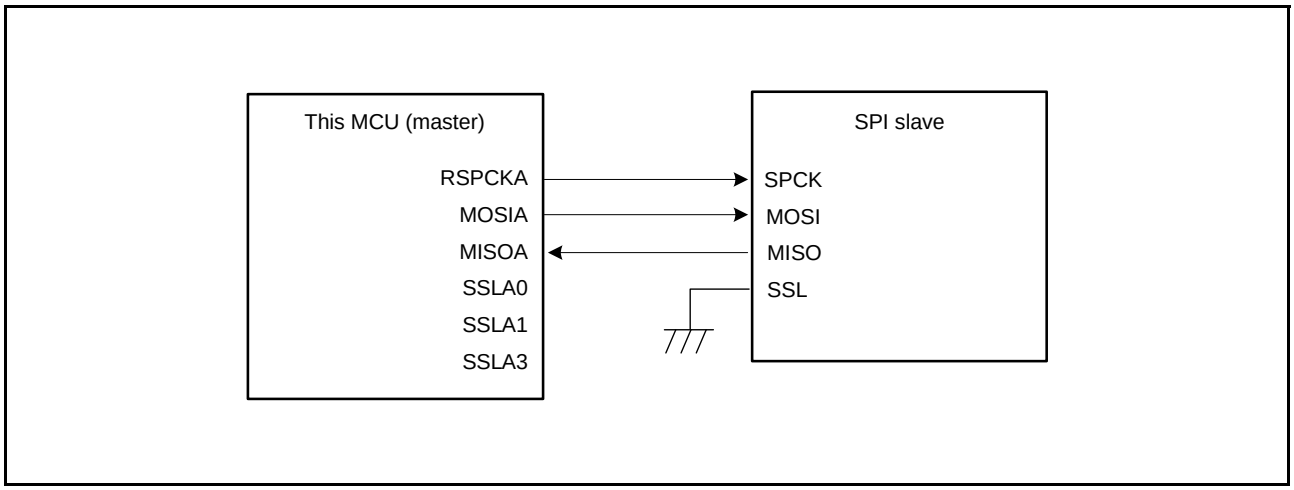



**Figure 38.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)**

### 38.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 38.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLA0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOA.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 38.7).

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

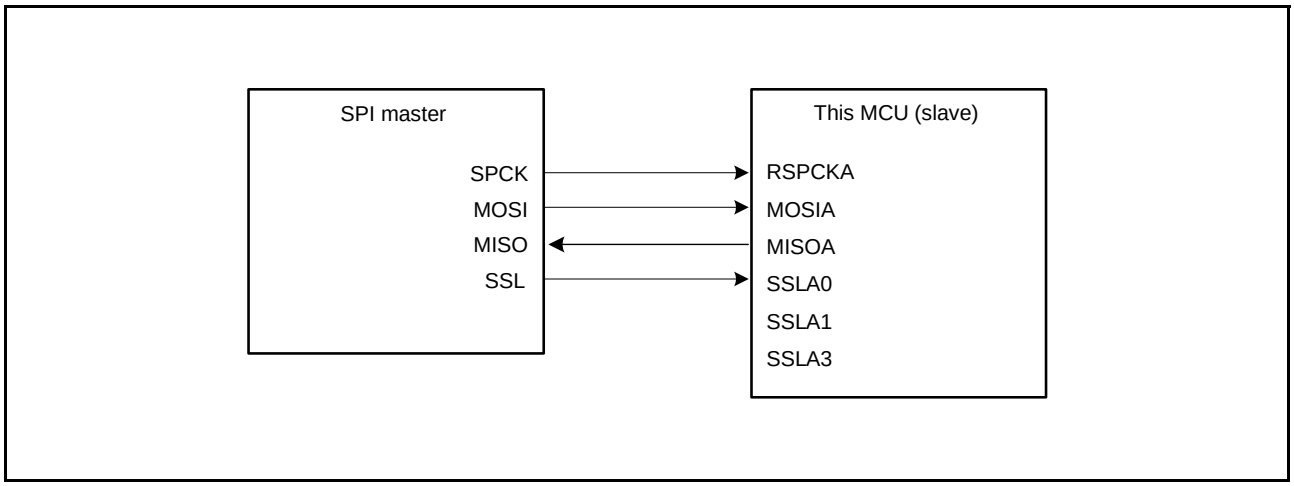



**Figure 38.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)**

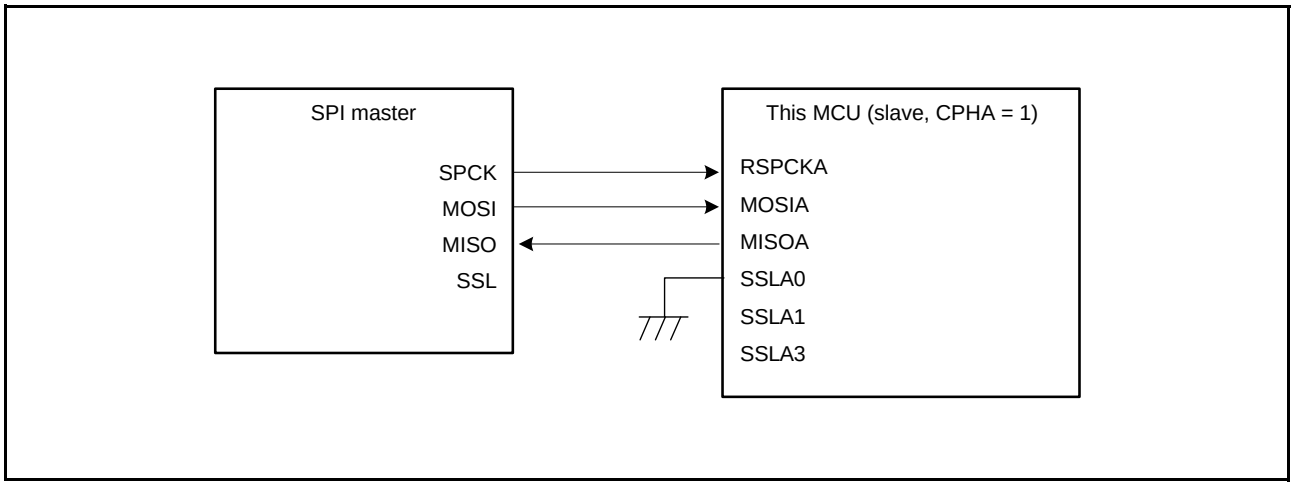



**Figure 38.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)**

#### 38.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 38.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 38.8, the RSPI system is comprised of this MCU (master) and three slaves (SPI slave 0 to SPI slave 2).

The RSPCKA and MOSIA outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 2. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOA input of this MCU (master). SSLA0, SSLA1, and SSLA3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 2, respectively.

This MCU (master) drives RSPCKA, MOSIA, SSLA0, SSLA1, and SSLA3. Of the SPI slave 0 to SPI slave 2, the slave that receives low-level input into the SSL input drives MISO.

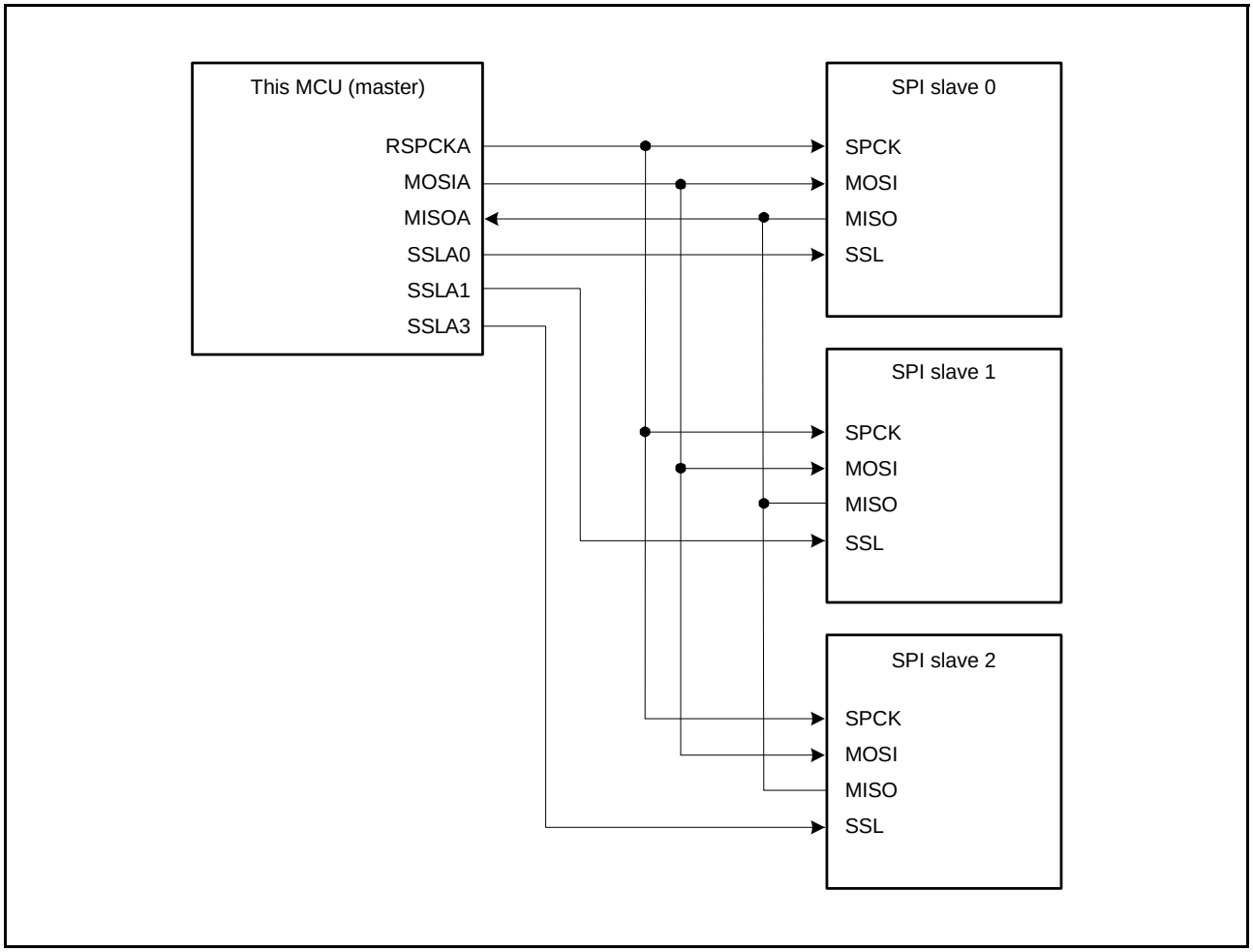



**Figure 38.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)**

#### 38.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 38.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 38.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y).
The SPCK and MOSI outputs of the SPI master are connected to the RSPCKA and MOSIA inputs of the MCUs (slave X and slave Y). The MISOA outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLA0 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

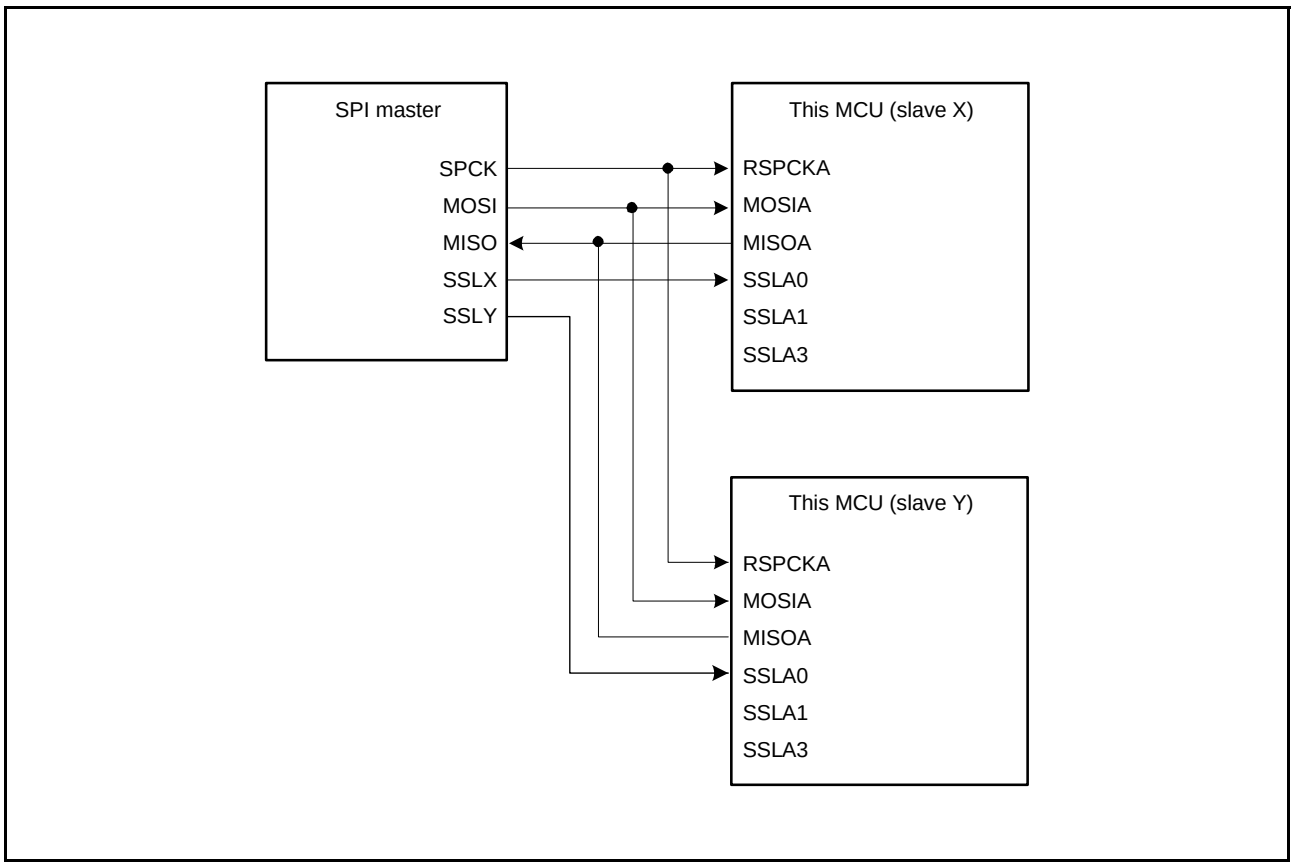



**Figure 38.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)**

### 38.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 38.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 38.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKA and MOSIA outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOA inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLA0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLA0 input of this MCU (master X). The SSLA1 and SSLA3 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2.

This MCU drives RSPCKA, MOSIA, SSLA1, and SSLA3 when the SSLA0 input level is high. When the SSLA0 input level is low, this MCU detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA3 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

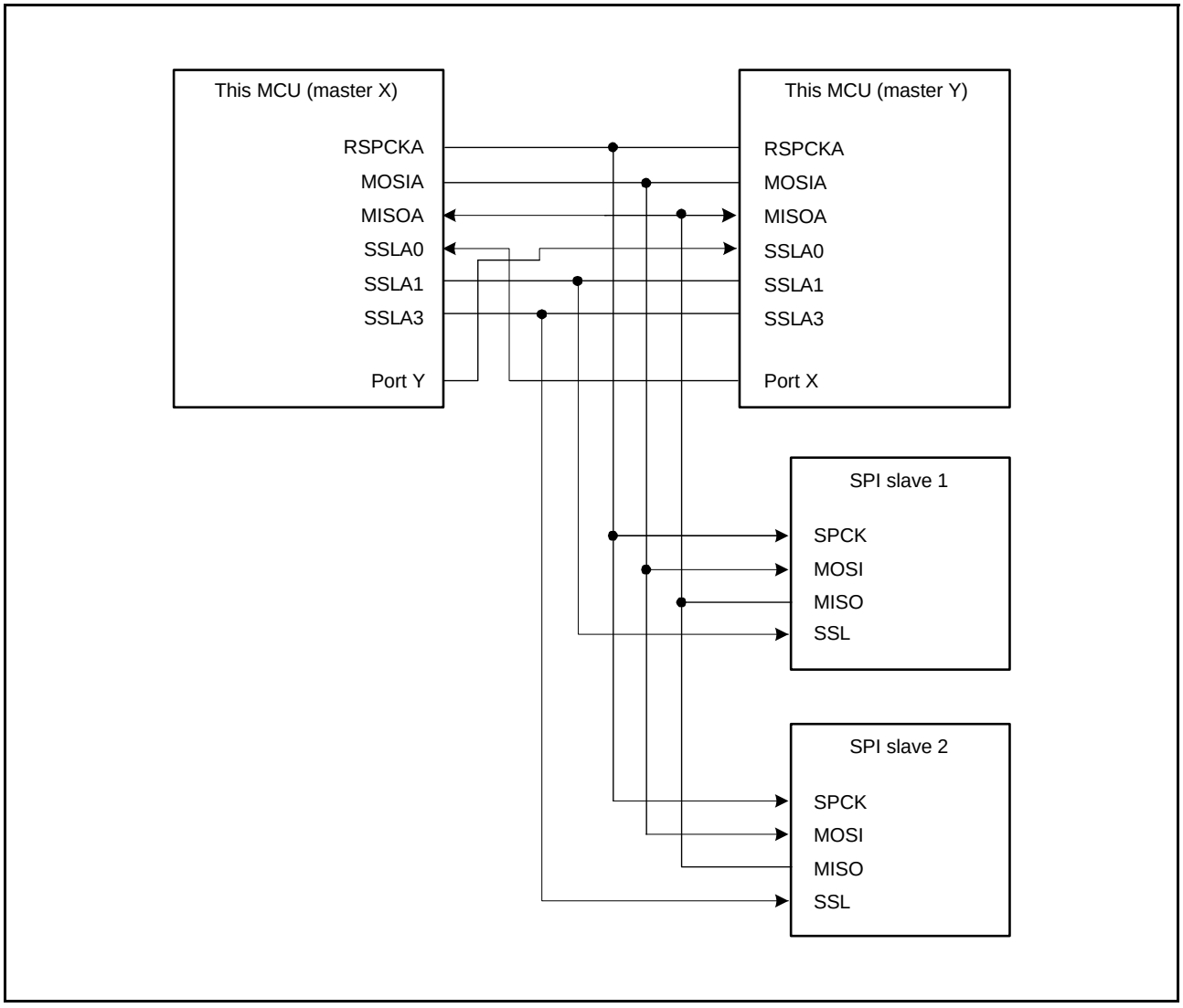



**Figure 38.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)**

### 38.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 38.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0, SSLA1, and SSLA3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

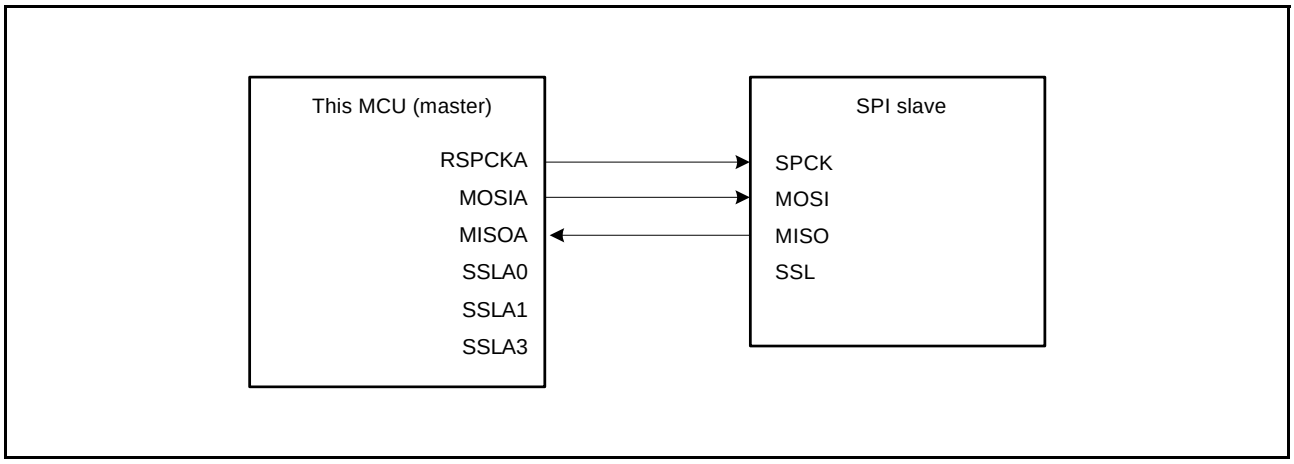



**Figure 38.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)**

### 38.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 38.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOA and the SPI master drives the SPCK and MOSI. In addition, SSLA0, SSLA1, and SSLA3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

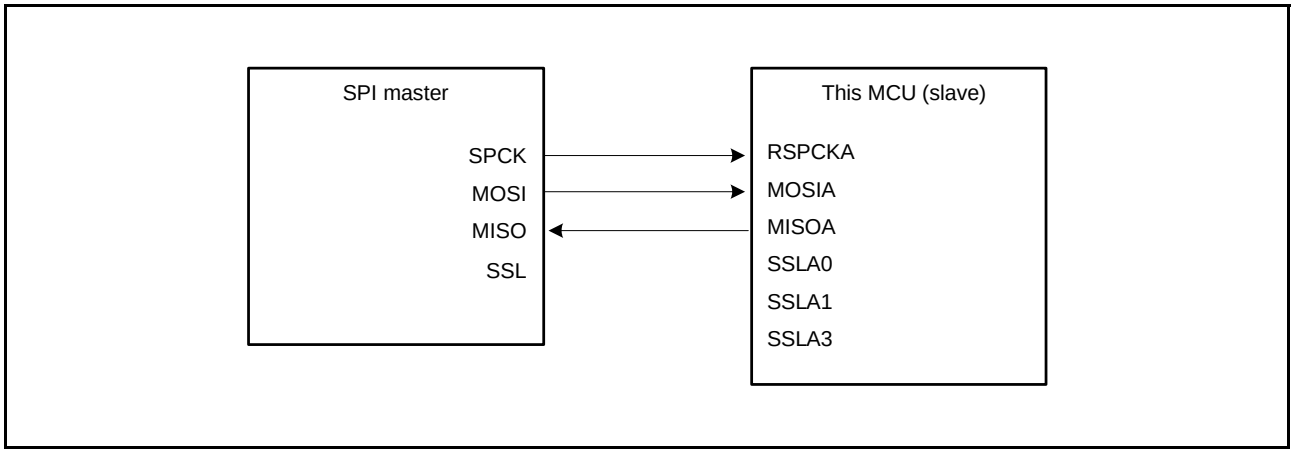



**Figure 38.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)**

### 38.3.4 Data Format

The RSPI's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transfer data.
The format of one frame of data before or after transfer is shown below.

#### (a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]).

#### (b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

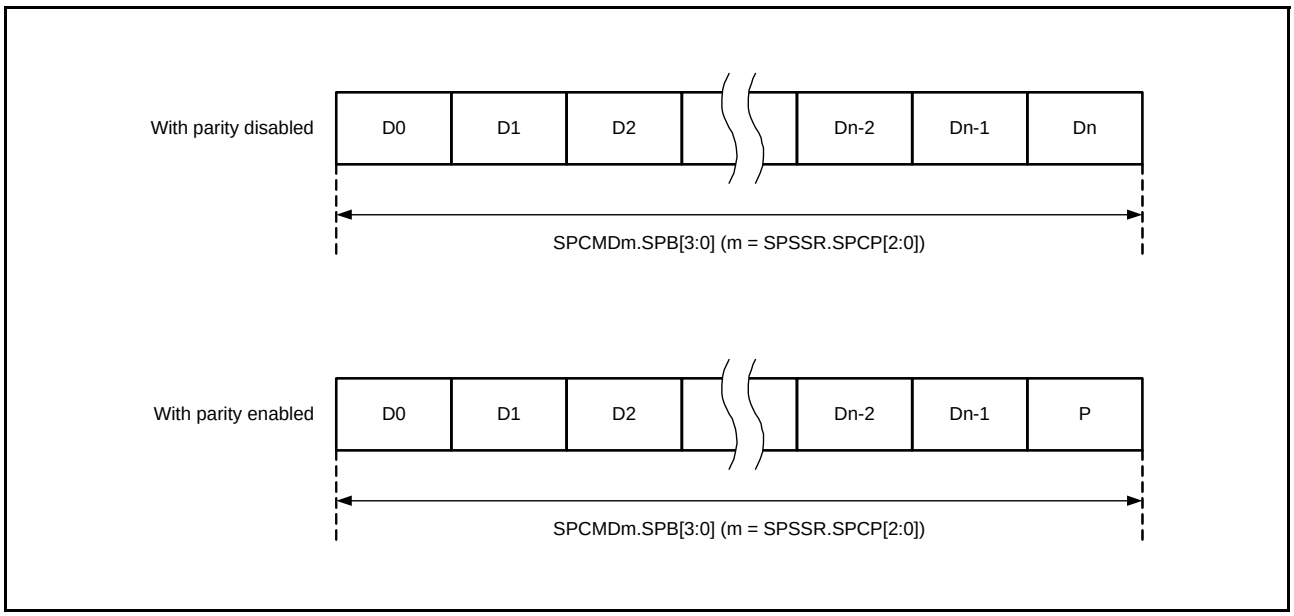



**Figure 38.13 Outline of the Data Format (with Parity Disabled/Enabled)**

### 38.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0)

When parity is disabled, data for transmission are copied to the shift register with no prior processing. A description of the connection between the RSPI data register (SPDR) and the shift register in terms of the combination of MSB or LSB first and data length is given below.

#### (1) MSB First Transfer (32-Bit Data)

Figure 38.14 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

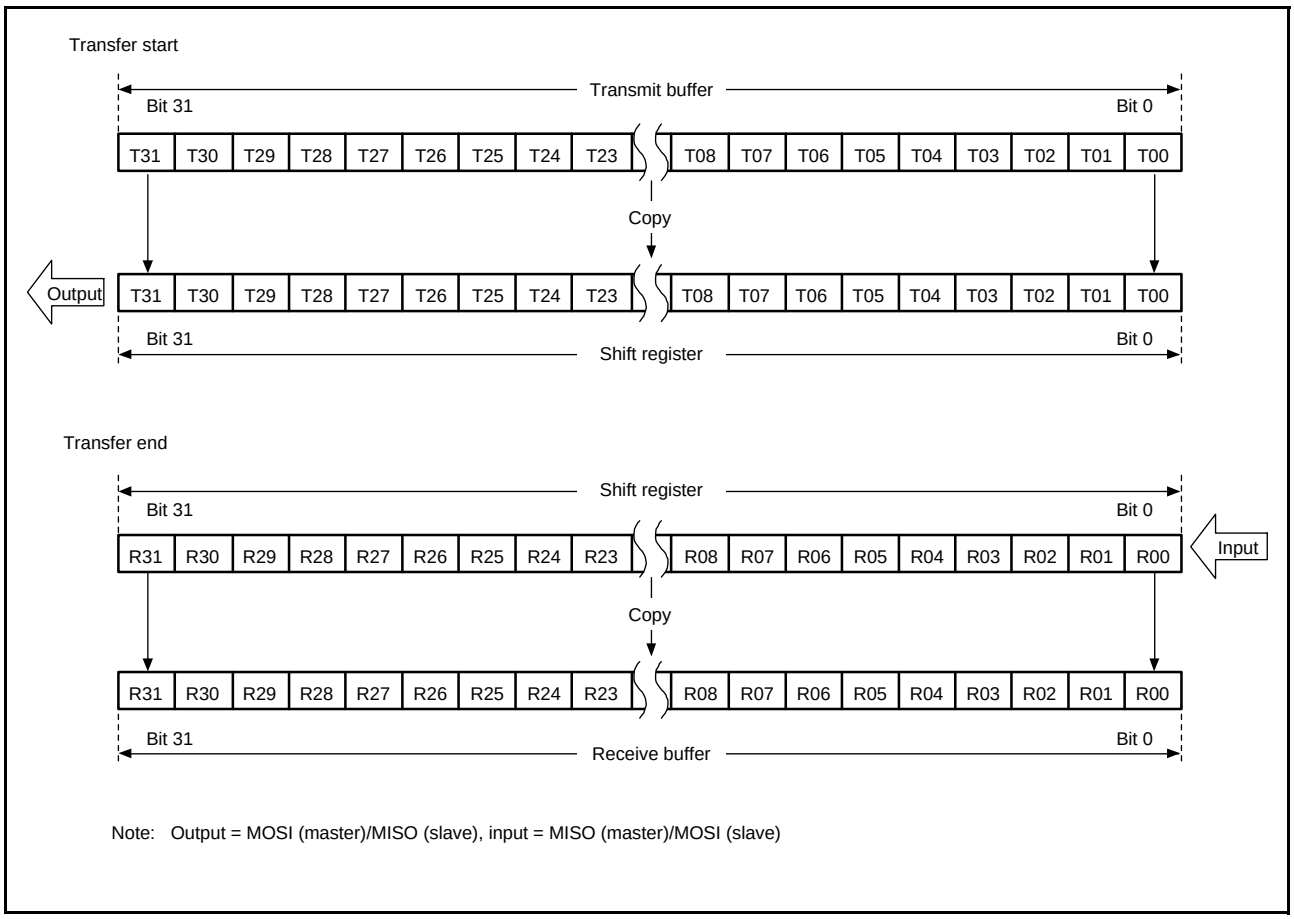



**Figure 38.14 MSB First Transfer (32-Bit Data, Parity Disabled)**

### (2) MSB First Transfer (24-Bit Data)

Figure 38.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.
Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

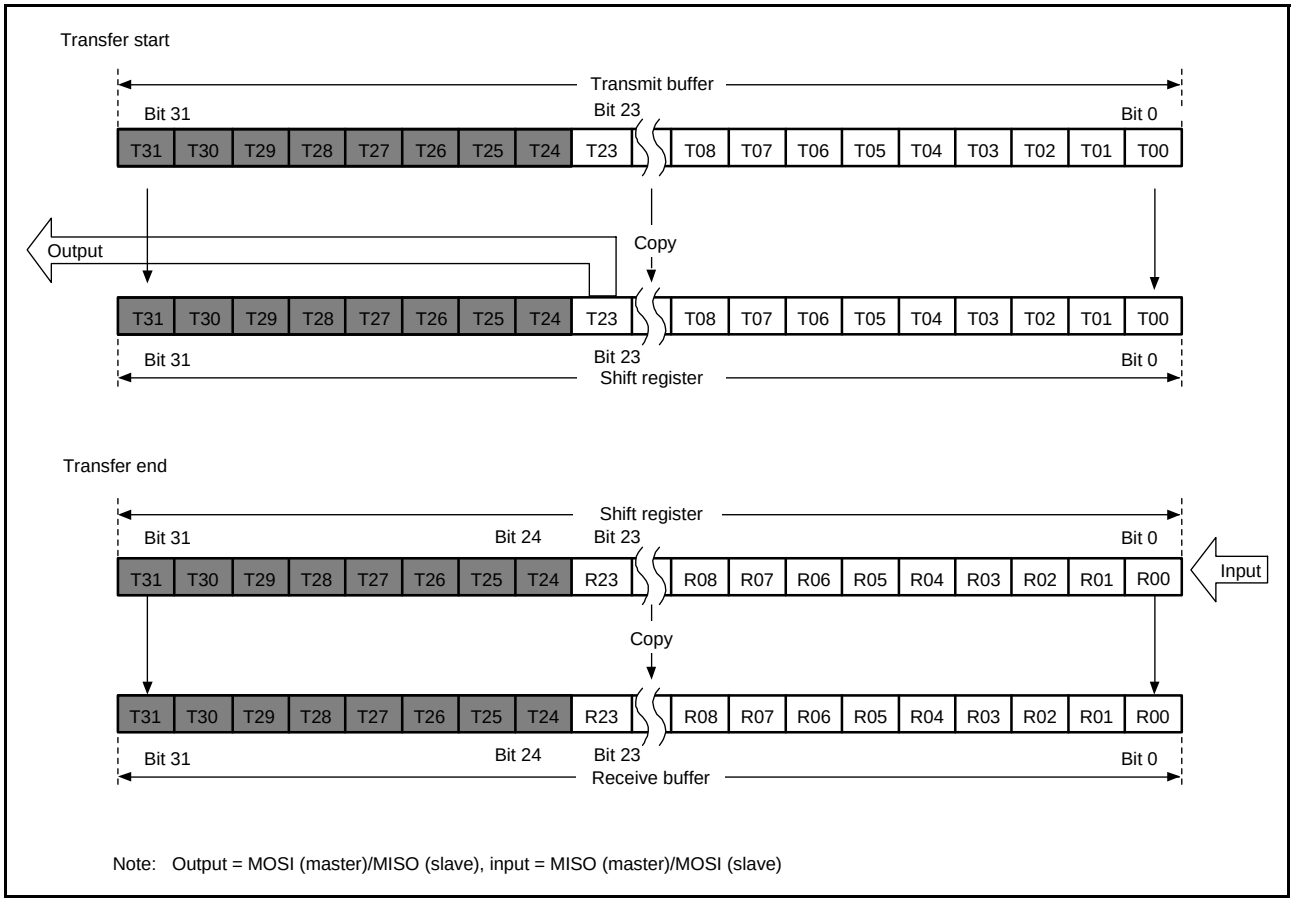



**Figure 38.15 MSB First Transfer (24-Bit Data, Parity Disabled)**

### (3) LSB First Transfer (32-Bit Data)

Figure 38.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

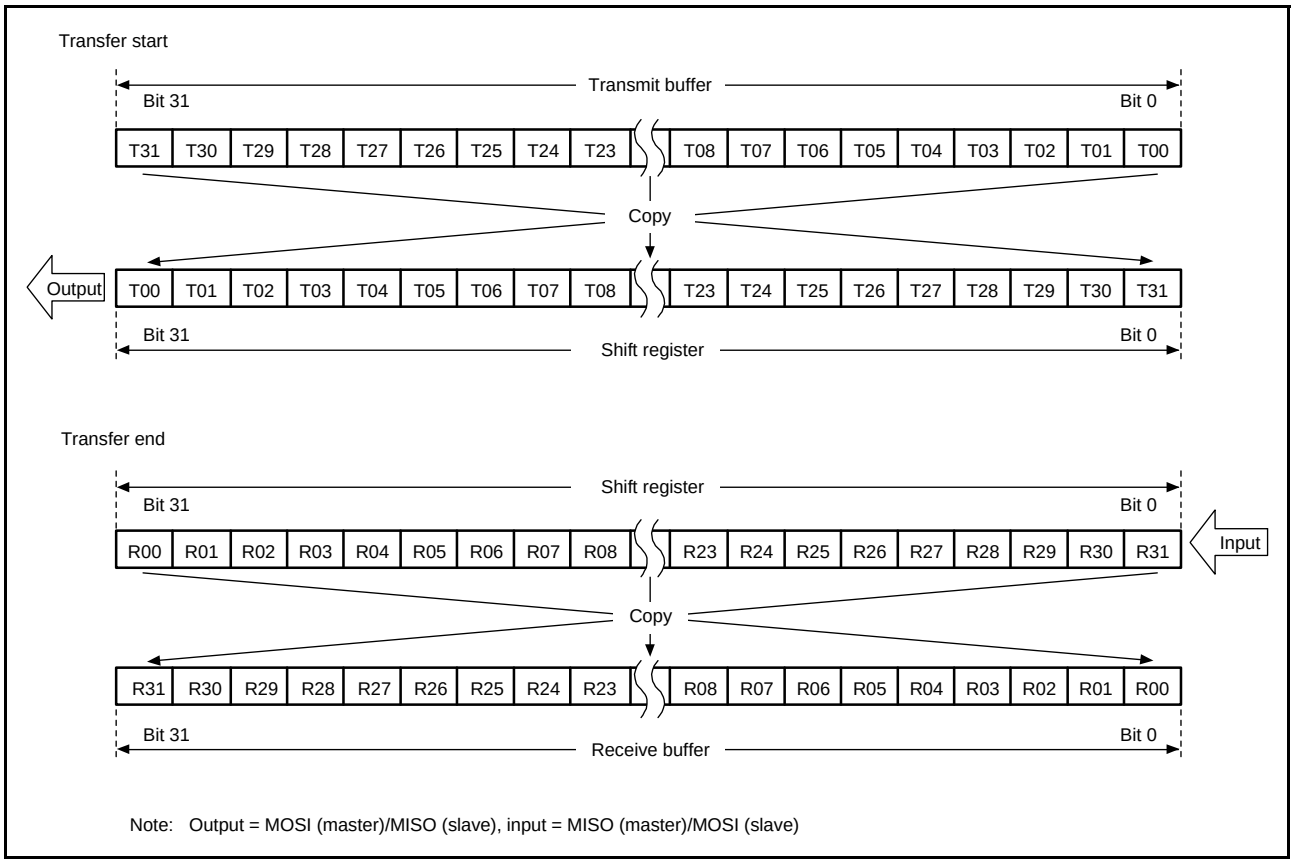



**Figure 38.16 LSB First Transfer (32-Bit Data, Parity Disabled)**

### (4) LSB First Transfer (24-Bit Data)

Figure 38.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.

Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

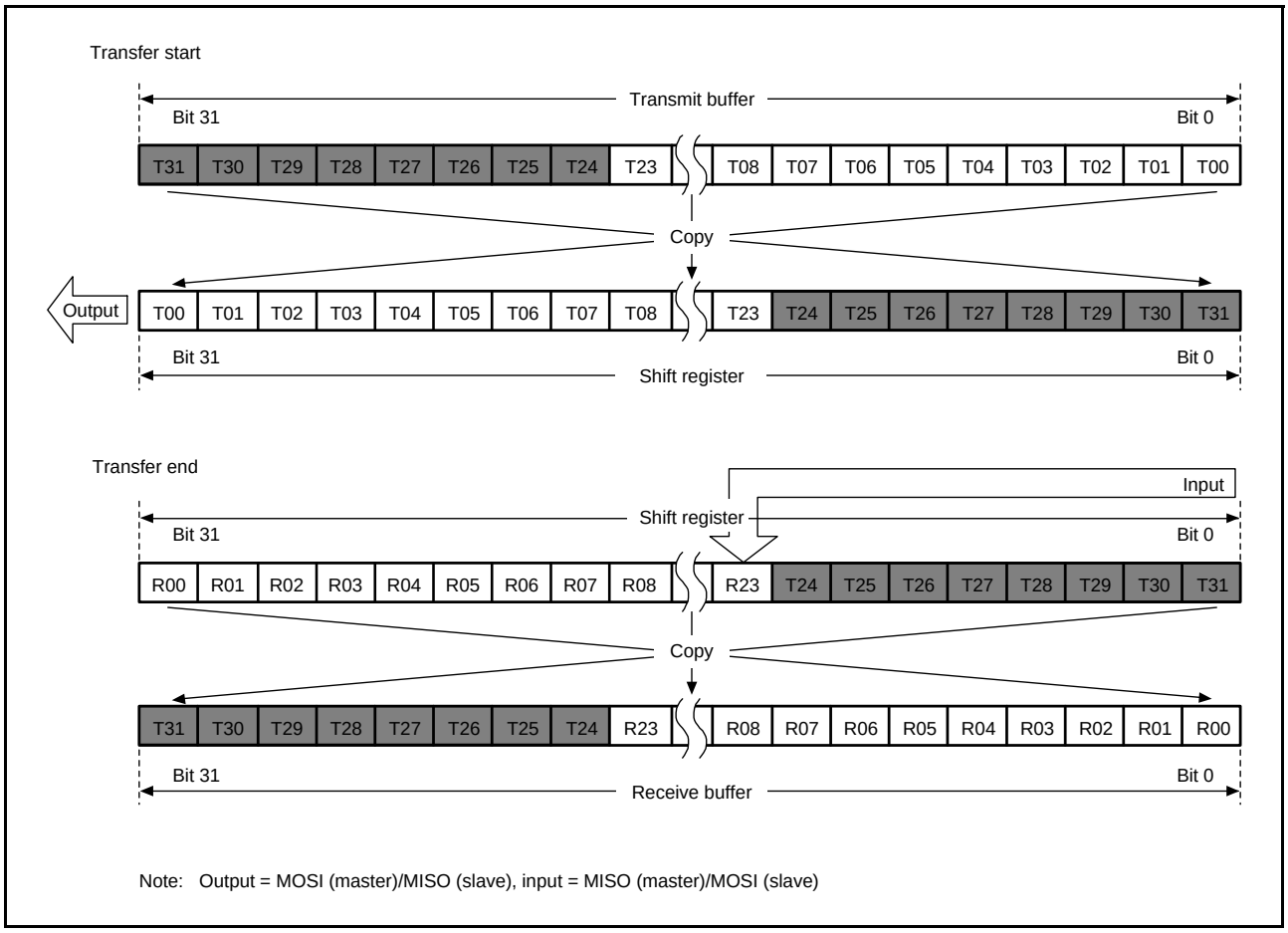



**Figure 38.17 LSB First Transfer (24-Bit Data, Parity Disabled)**

### 38.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

#### (1) MSB First Transfer (32-Bit Data)

Figure 38.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

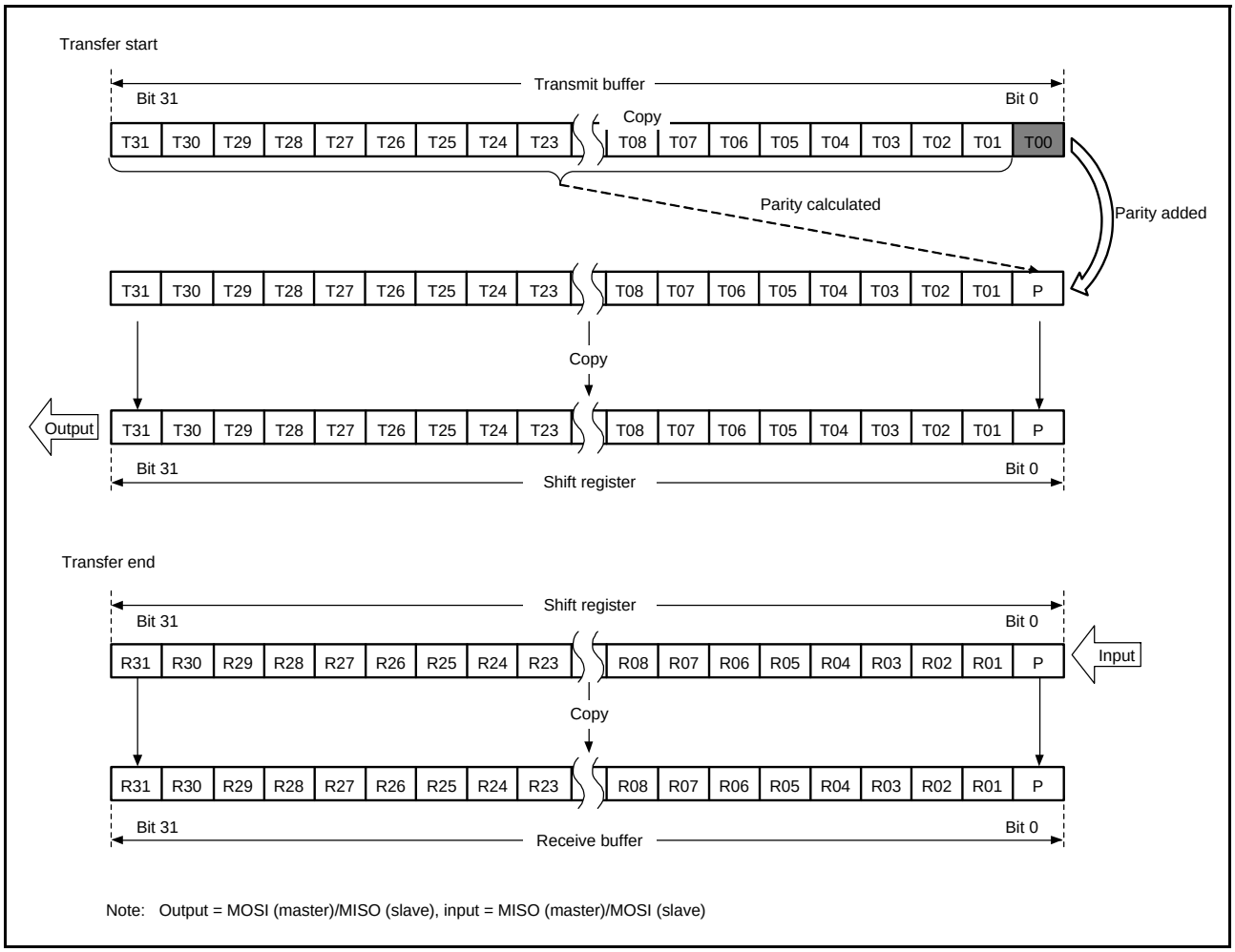



**Figure 38.18 MSB First Transfer (32-Bit Data, Parity Enabled)**

### (2) MSB First Transfer (24-Bit Data)

Figure 38.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

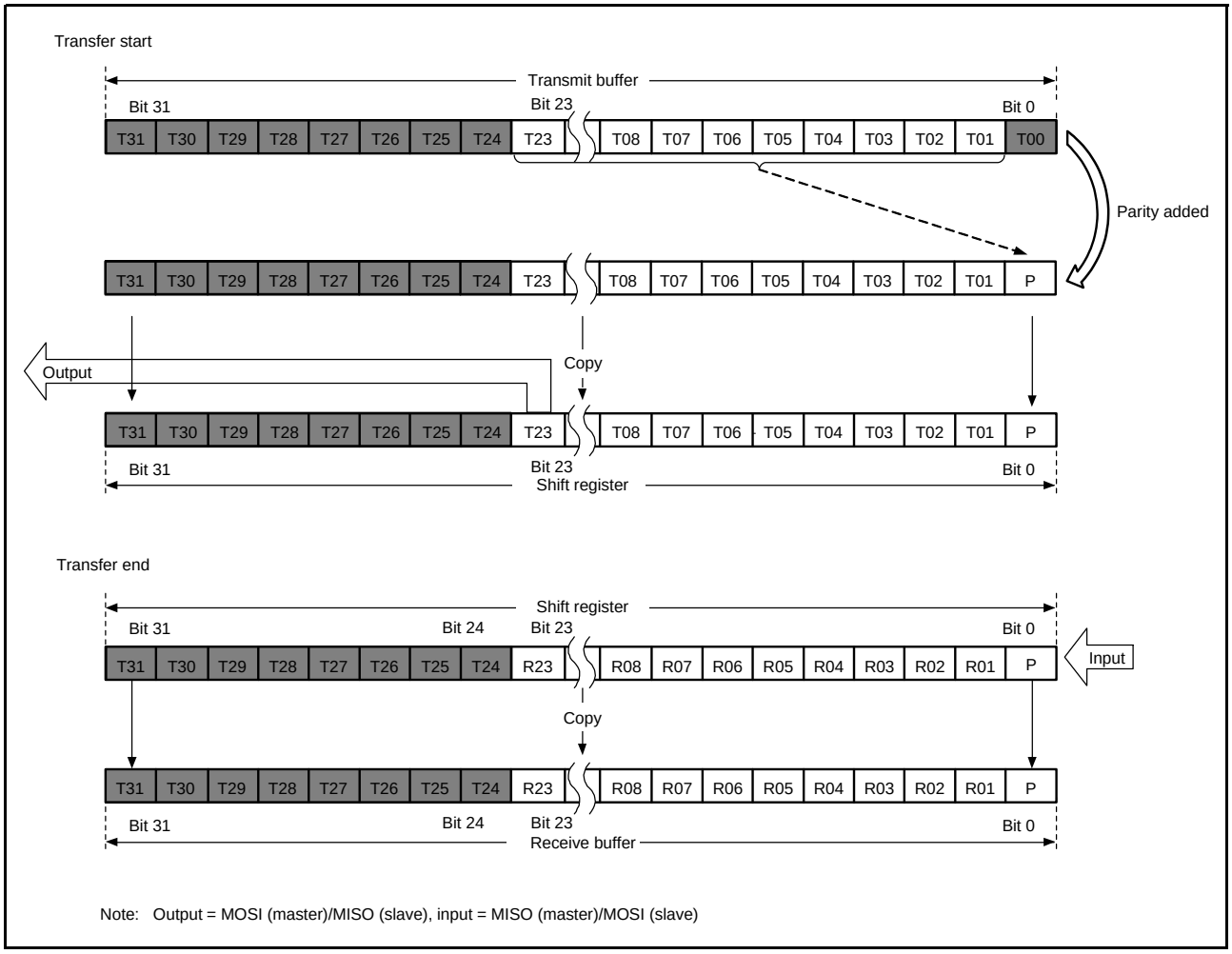



**Figure 38.19 MSB First Transfer (24-Bit Data, Parity Enabled)**

### (3) LSB First Transfer (32-Bit Data)

Figure 38.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity.

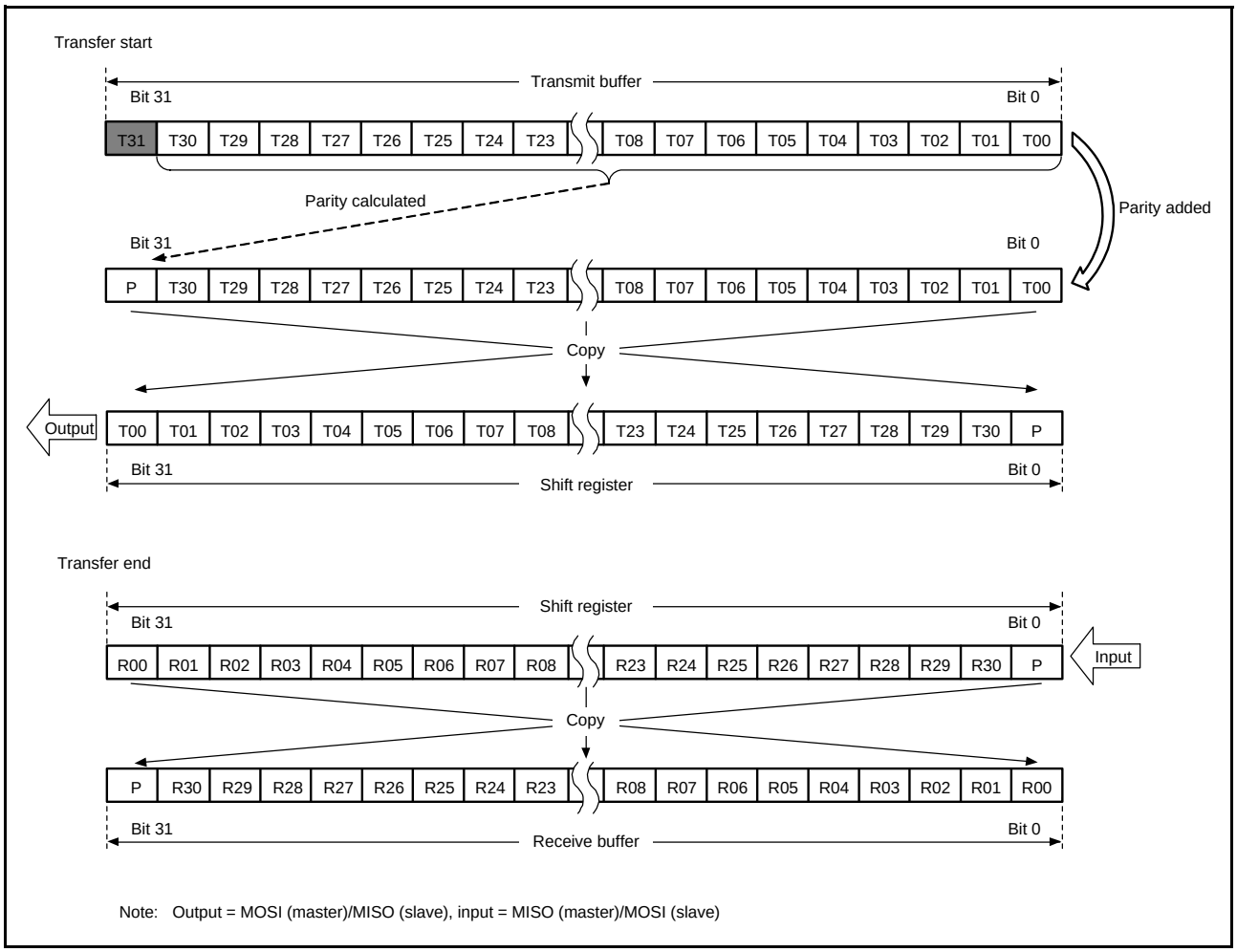



**Figure 38.20 LSB First Transfer (32-Bit Data, Parity Enabled)**

### (4) LSB First Transfer (24-Bit Data)

Figure 38.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

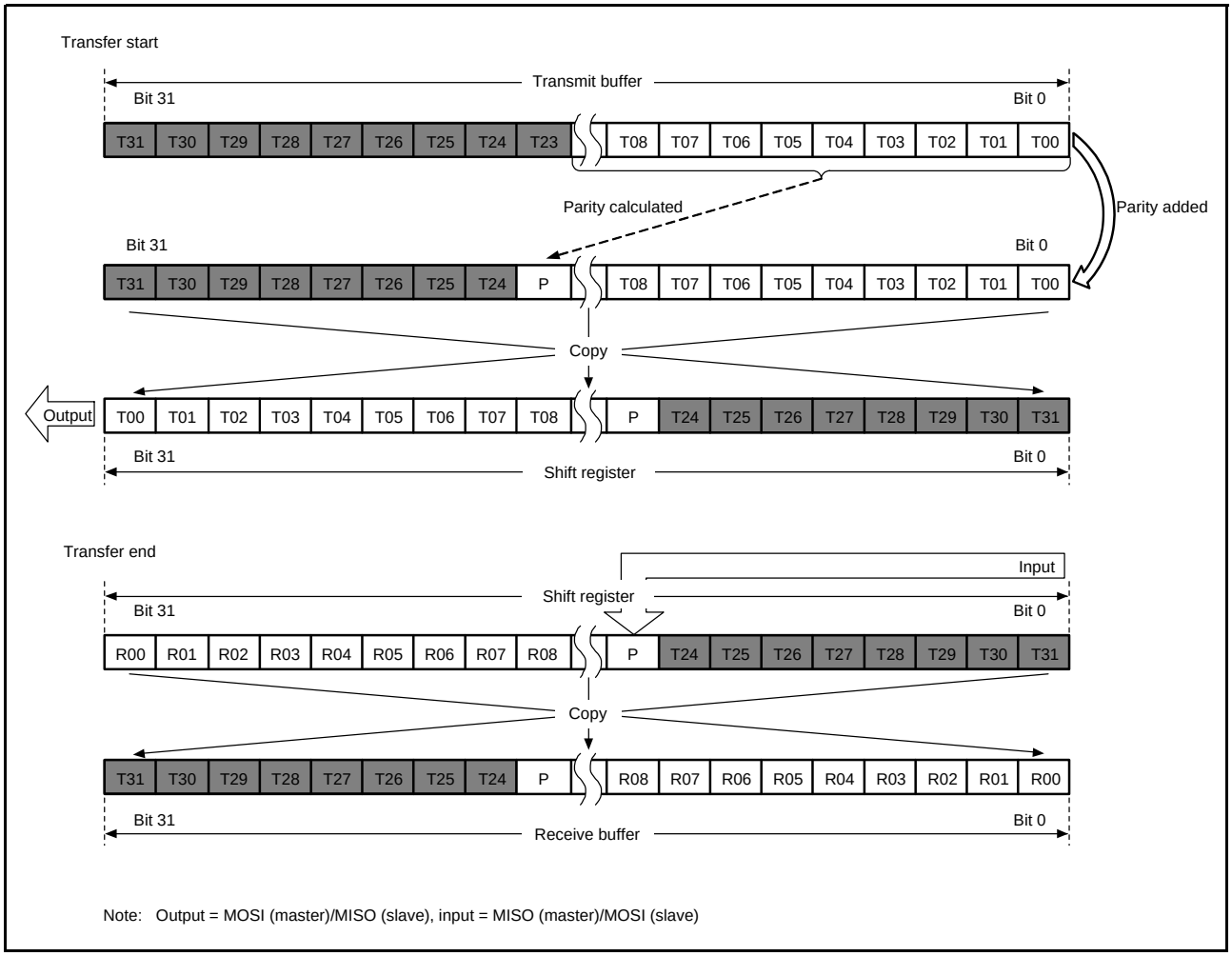



**Figure 38.21 LSB First Transfer (24-Bit Data, Parity Enabled)**

### 38.3.5 Transfer Format

#### 38.3.5.1 CPHA = 0

Figure 38.22 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 38.22, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI settings. For details, refer to section 38.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 38.3.10.1, Master Mode Operation.

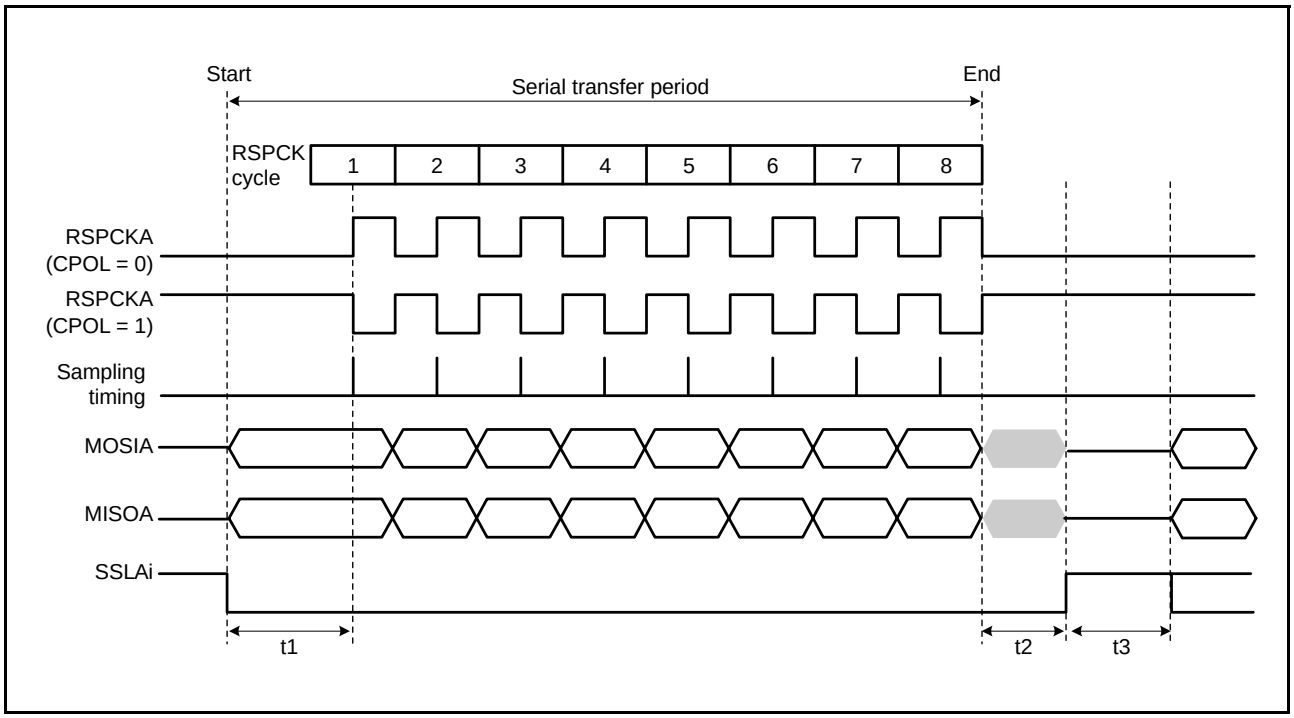



**Figure 38.22 RSPI Transfer Format (CPHA = 0)**

### 38.3.5.2 CPHA = 1

Figure 38.23 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 38.23, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 38.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 38.3.10.1, Master Mode Operation.

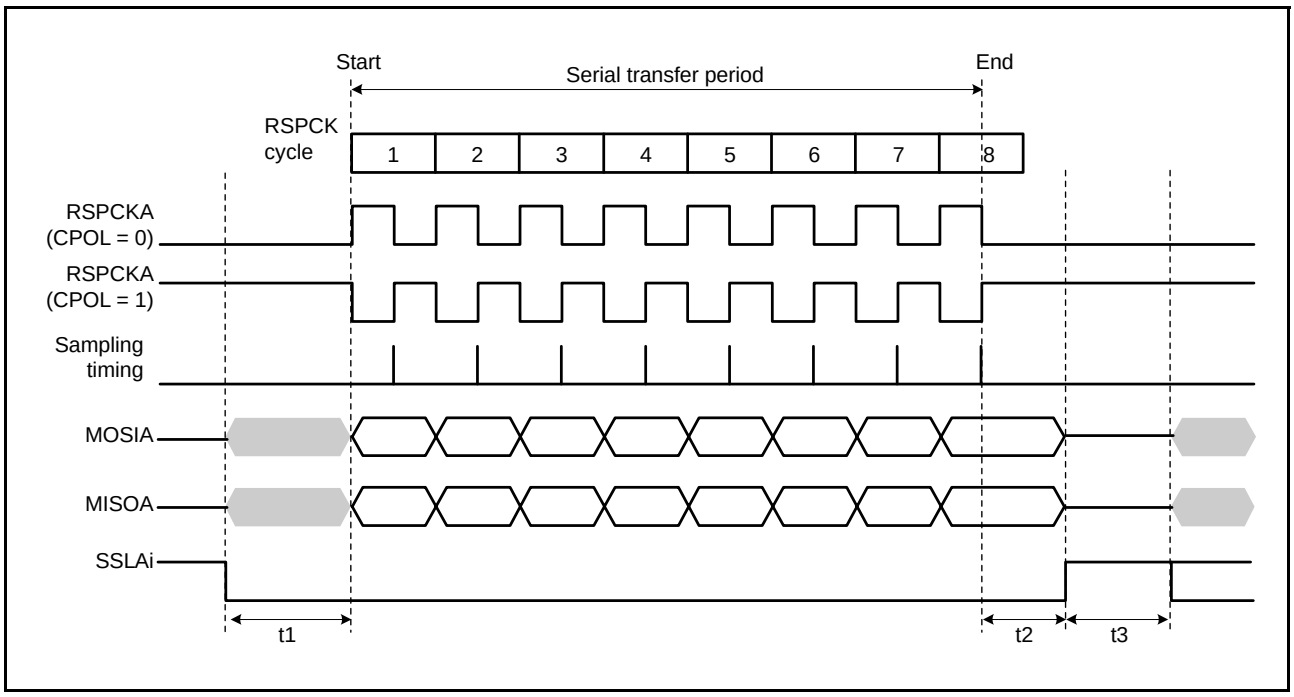



**Figure 38.23 RSPI Transfer Format (CPHA = 1)**

### 38.3.6 Communications Operating Mode

Full-duplex synchronous serial communications or transmit operations only can be selected by the communications operating mode select bit (SPCR.TXMD). The SPDR access shown in Figure 38.24 and Figure 38.25 indicate the condition of access to the SPDR register, where W denotes a write cycle.

#### 38.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 38.24 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 38.24, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

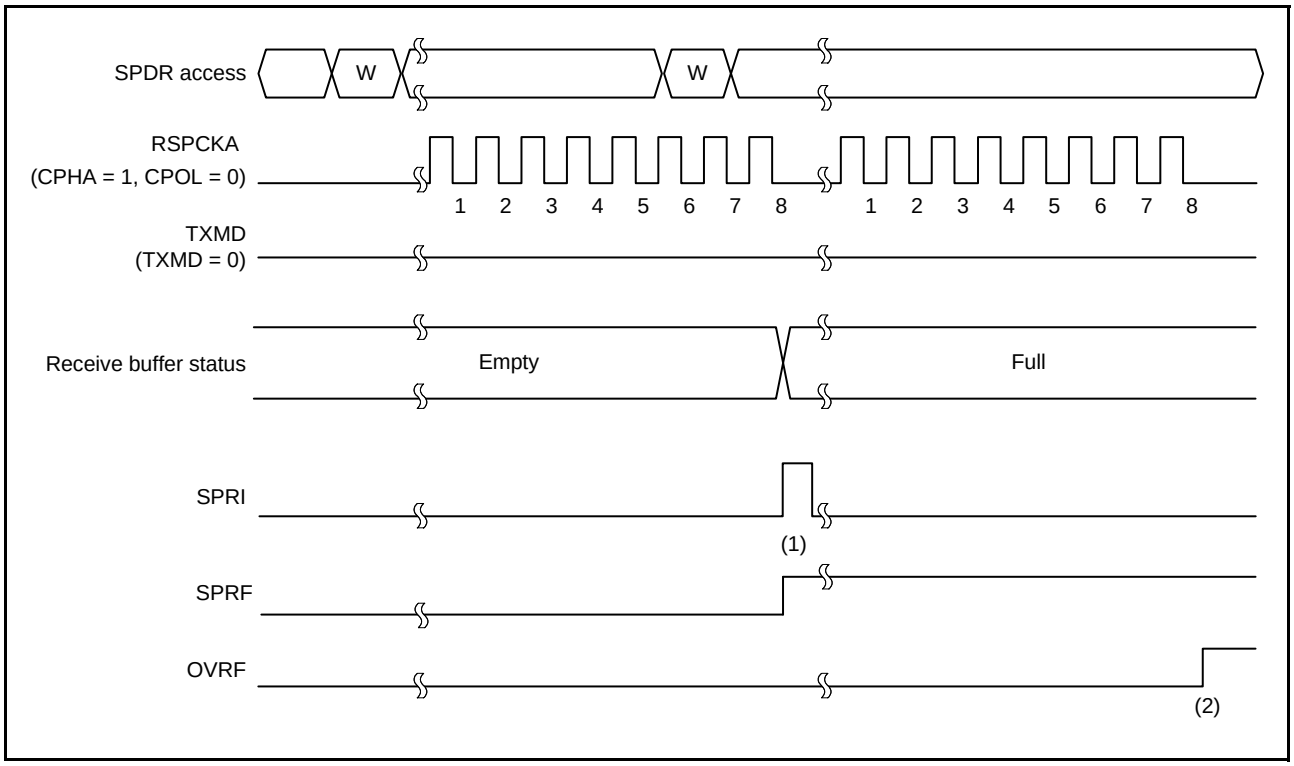



**Figure 38.24 Operation Example of SPCR.TXMD = 0**

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends with the receive buffer of SPDR empty, the RSPI generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of SPDR holding data that was received in the previous serial transfer, the RSPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

When full-duplex synchronous serial communications (SPCR.TXMD = 0) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

### 38.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 38.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 38.25, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

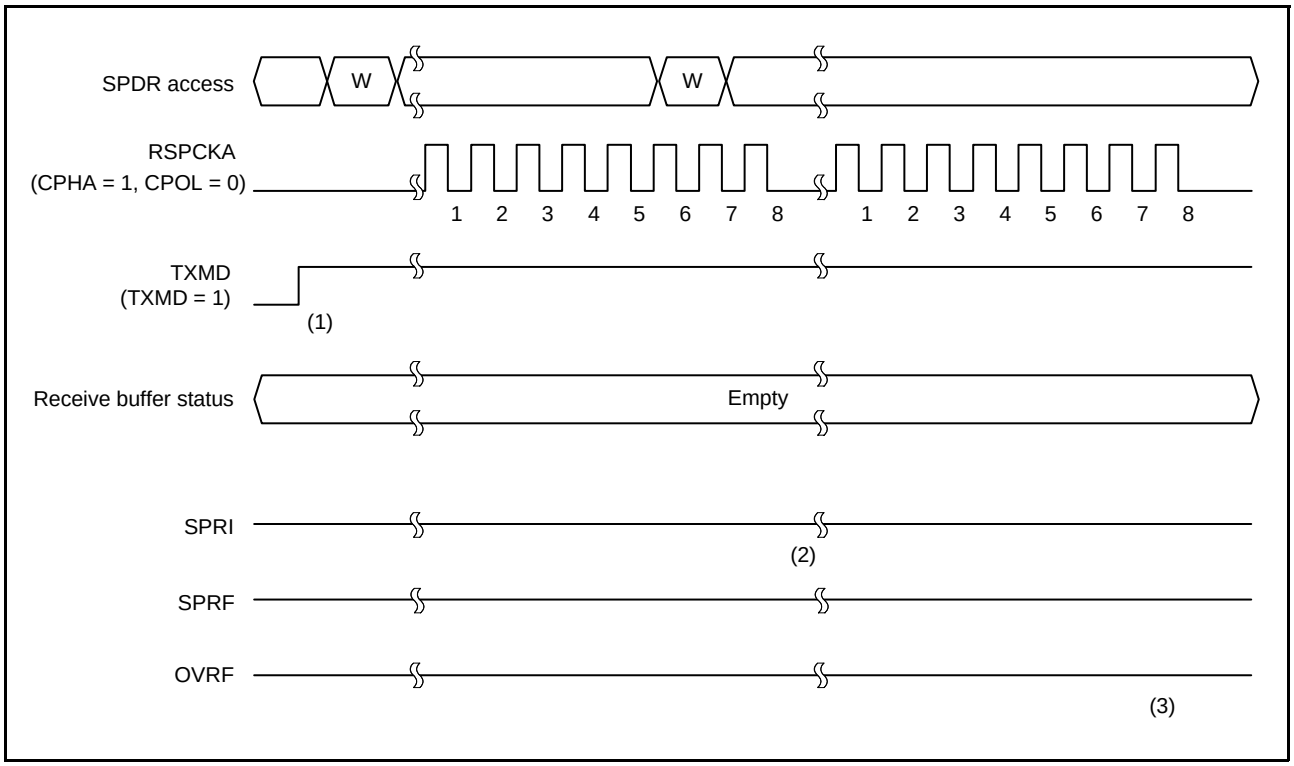



**Figure 38.25 Operation Example of SPCR.TXMD = 1**

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit operations only (SPCR.TXMD = 1).

(2) When a serial transfer ends with the receive buffer of SPDR empty, if the mode of transmit operations only is selected (SPCR.TXMD = 1), the SPRF flag remains 0 and the RSPI does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When performing transmit operations only (SPCR.TXMD = 1), the RSPI transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

### 38.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 38.26 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 38.26 indicates the condition of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the example in Figure 38.26, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

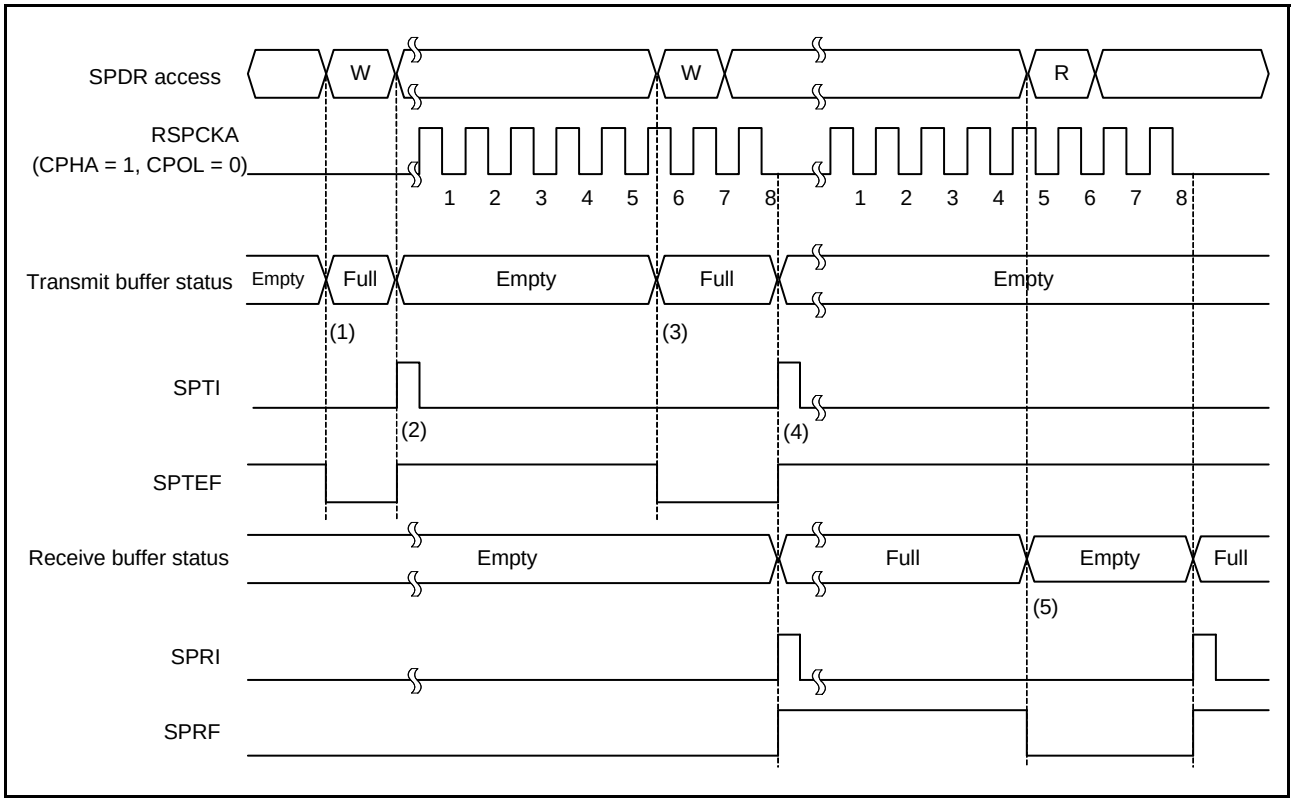



**Figure 38.26 Operation Example of SPTI and SPRI Interrupts**

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

(1) When transmit data is written to SPDR when the transmit buffer of SPDR is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer and sets the SPSR.SPTEF flag to 0.

(2) If the shift register is empty, the RSPI copies the data from the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPI. For details, refer to section 38.3.10, SPI Operation, and section 38.3.11, Clock Synchronous Operation.

(3) When transmit data is written to SPDR in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer and the SPSR.SPTEF flag becomes 0. Because the data being transmitted is stored in the shift register, the RSPI does not copy the data from the transmit buffer to the shift register.

(4) When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI copies the receive data from the shift register to the receive buffer, generates a receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of serial transfer, when the transmit buffer had been full before the serial transfer ended, the RSPI sets the SPSR.SPTEF flag to 1 and copies the data from the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer, the RSPI determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.

(5) When SPDR is read in the receive buffer full interrupt routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. When the receive data is read, the SPRF flag becomes 0.

If transmit data is written to SPDR while the transmit buffer holds data that has not yet been transmitted (the SPTEF flag is 0), the RSPI does not update the data in the transmit buffer. Transmit data should be written to SPDR in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.
When setting the SPCR.SPE bit to 0 (RSPI disabled), the SPCR.SPTIE bit should also be set to 0. Otherwise (if the SPCR.SPE bit is 0 and the SPCR.SPTIE is 1), a transmit buffer empty interrupt request will occur.
When serial transfer ends with the receive buffer being full (the SPRF flag is 1), the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 38.3.8, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an RSPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.
Transmit and receive interrupts or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. Refer to section 15, Interrupt Controller (ICUb), for the interrupt vector numbers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

### 38.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is transmitted, and the received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error, parity error, or mode fault error. Table 38.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

**Table 38.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function**

| | Occurrence Condition | RSPI Operation | Error Detection |
|---|---|---|---|
| 1 | SPDR is written when the transmit buffer is full. | • The contents of the transmit buffer are kept.<br>• Missing write data. | None |
| 2 | SPDR is read when the receive buffer is empty. | Data received previously is output to the bus. | None |
| 3 | Serial transfer is started in slave mode when transmit data is still not loaded on the shift register. | Data received in previous serial transfer is transmitted. | None |
| 4 | Serial transfer terminates when the receive buffer is full. | • The contents of the receive buffer are kept.<br>• Missing receive data. | Overrun error |
| 5 | An incorrect parity bit is received when performing full-duplex synchronous serial communications with the parity function enabled. | The parity error flag is asserted. | Parity error |
| 6 | The SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode. | • Driving of the RSPCKA, MOSIA, SSLA1, SSLA3 output signals is stopped.<br>• RSPI function is disabled. | Mode fault error |
| 7 | The SSLA0 input signal is asserted during serial transfer in multi-master mode. | • Serial transfer is suspended.<br>• Missing transmit/receive data.<br>• Driving of the RSPCKA, MOSIA, SSLA1, SSLA3 output signals is stopped.<br>• RSPI function is disabled. | Mode fault error |
| 8 | The SSLA0 input signal is negated during serial transfer in slave mode. | • Serial transfer is suspended.<br>• Missing transmit/receive data.<br>• Driving of the MISOA output signal is stopped.<br>• RSPI function is disabled. | Mode fault error |

On operation 1 described in Table 38.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPI does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when an RSPI receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

Similarly, the RSPI does not detect an error on operation 3. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in 3 as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read (if SPDR is not read before the end of the serial transfer, an overrun error may occur).

An overrun error shown in 4 is described in section 38.3.8.1, Overrun Error. A parity error shown in 5 is described in section 38.3.8.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 38.3.8.3, Mode Fault Error. For the transmit and receive interrupts, refer to section 38.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

### 38.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read SPSR with the OVRF flag set to 1.
Figure 38.27 shows an example of operations of the SPRF and OVRF flags. The SPSR and SPDR accesses shown in Figure 38.27 indicate the condition of accesses to SPSR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 38.27, the RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

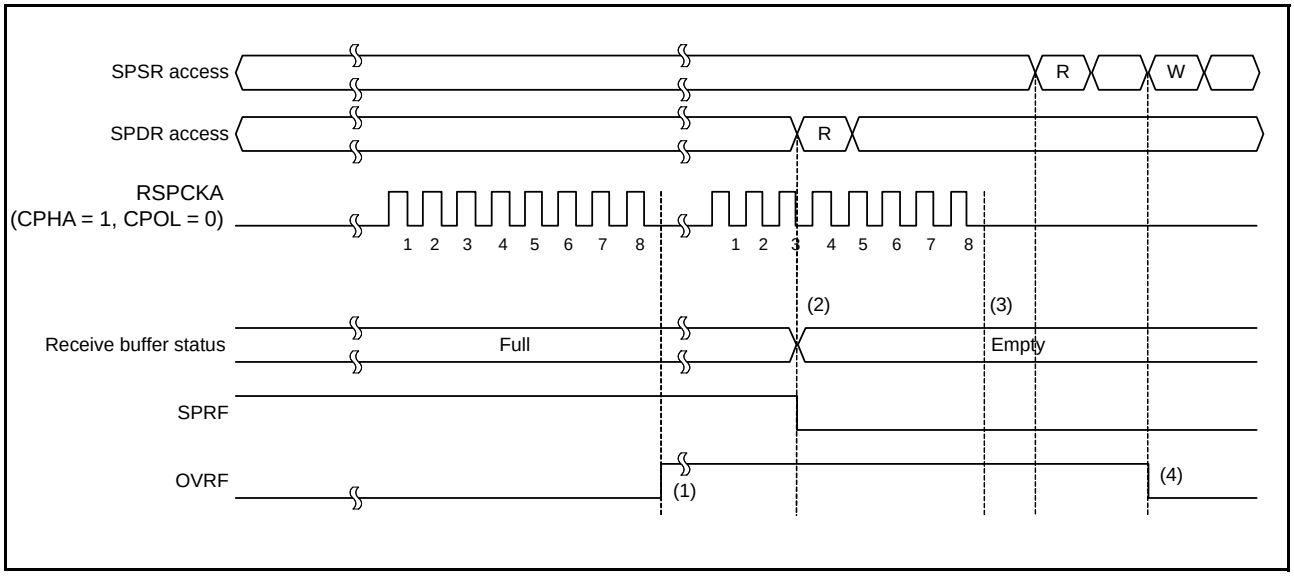



**Figure 38.27 Operation Example of SPRF and OVRF Flags**

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

(1) If a serial transfer terminates with the receive buffer full (the SPRF flag is 1), the RSPI detects an overrun error, and sets the OVRF flag to 1. The RSPI does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When SPDR is read, the RSPI outputs the data in the receive buffer. At this time the SPRF flag becomes 0. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPI does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.

(4) If 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 38.28 and Figure 38.29 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

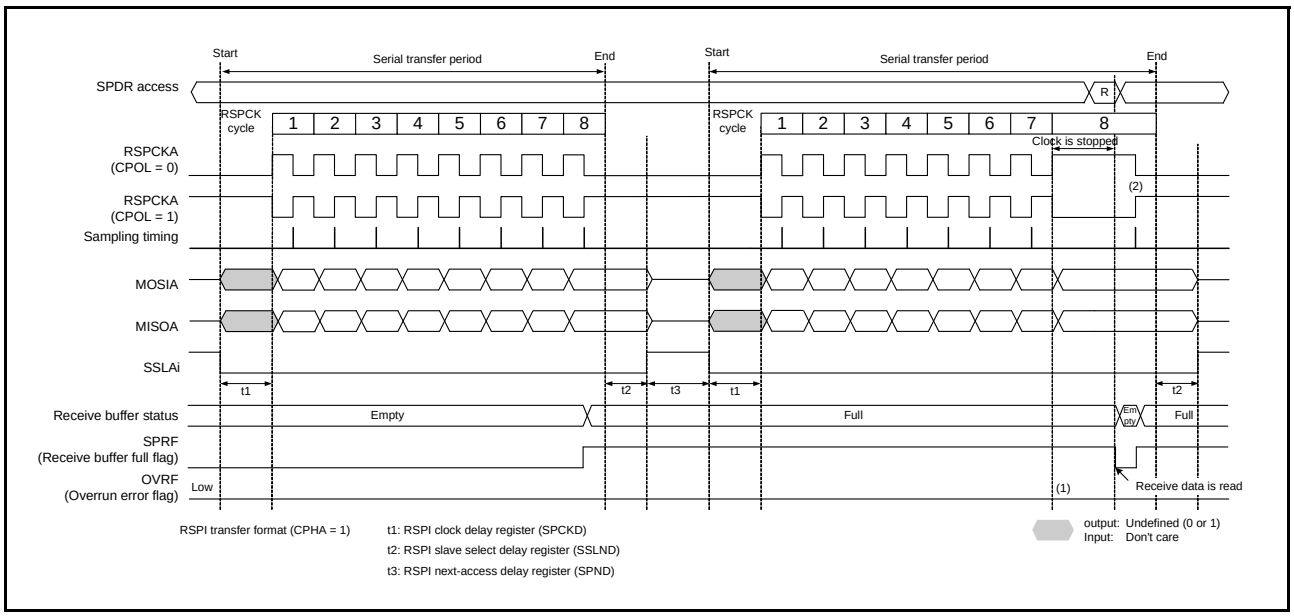



**Figure 38.28 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1)**

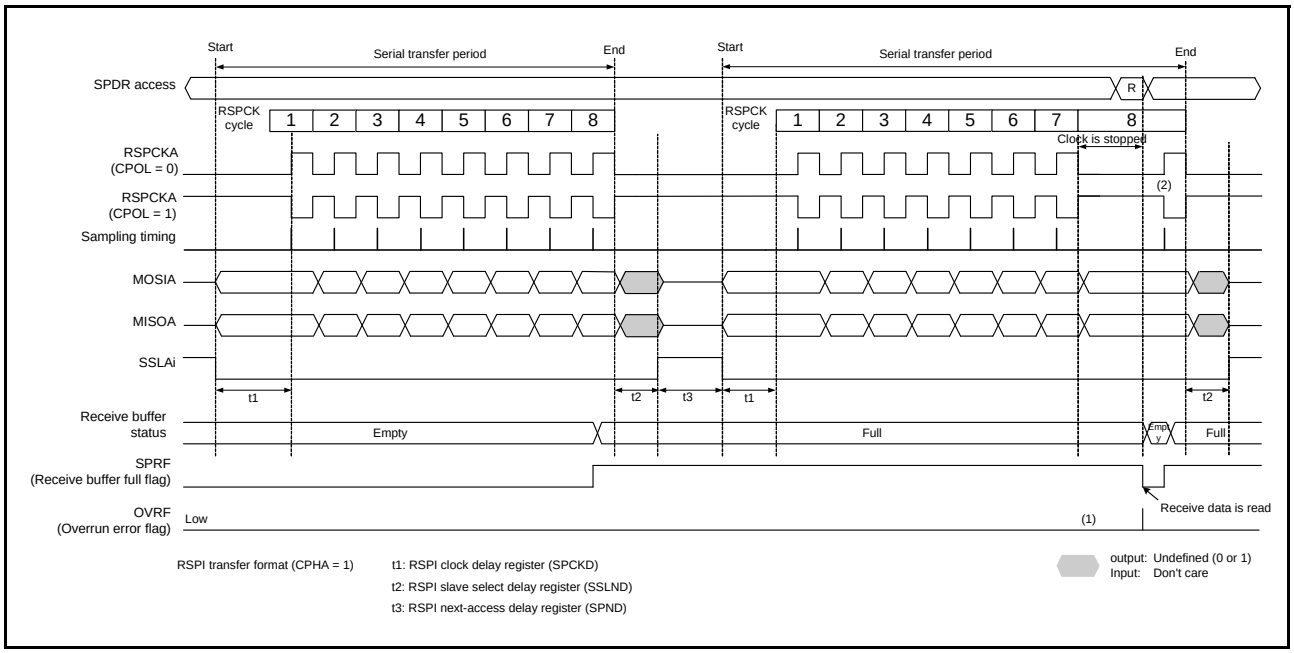



**Figure 38.29 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0)**

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after the SPRF flag becomes 0).

### 38.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the RSPI checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPI sets the SPSR.PERF flag to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after SPSR register is read with the PERF flag set to 1.
Figure 38.30 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 38.30 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 38.30, full-duplex synchronous serial communications is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

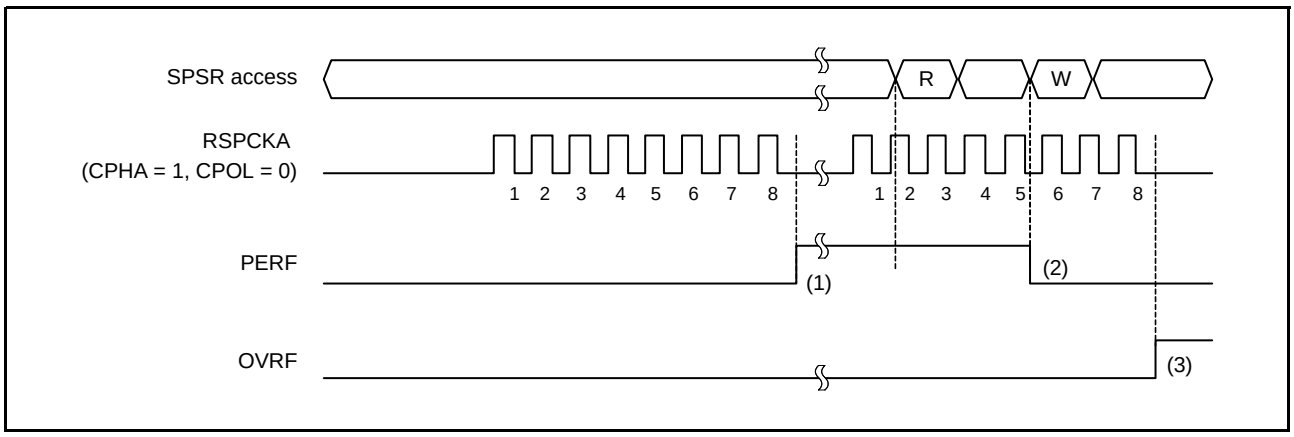



**Figure 38.30 Operation Example of PERF Flag**

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

(1) If a serial transfer terminates with the RSPI not detecting an overrun error, the RSPI copies the data in the shift register to the receive buffer. The RSPI judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) If 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.

(3) When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an RSPI error interrupt and reading the SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of parity errors, such as reading SPSR. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

#### 38.3.8.3 Mode Fault Error

The RSPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input with respect to the SSLA0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLA0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit of the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLA0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (refer to section 38.3.9, Initializing RSPI). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. Detecting mode fault errors without utilizing the RSPI error interrupt requires polling of SPSR. When using the RSPI in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of a mode fault error, set the MODF flag to 0.

### 38.3.9 Initializing RSPI

If 0 is written to the SPCR.SPE bit or the RSPI sets the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

#### 38.3.9.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPI
- Initializing the transmit buffer of the RSPI (Set the SPTEF flag to 1)

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.SPRF, SPSR.OVRF, SPSR.MODF, and SPSR.PERF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.
The transmit buffer is initialized to an empty state (the SPTEF flag is 1). Therefore, if the SPCR.SPTIE bit is set to 1 after RSPI initialization, a transmit buffer empty interrupt is generated. When the RSPI is initialized, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

#### 38.3.9.2 System Reset

The initialization by a system reset completely initializes the RSPI through the initialization of all bits for controlling the RSPI, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 38.3.9.1, Initialization by Clearing the SPE Bit.

## 38.3.10 SPI Operation

### 38.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 38.3.8, Error Detection). When operating in single-master mode, the RSPI does not detect mode fault errors whereas the RSPI running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-master mode and multi-master mode.

#### (1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) when data is written to the RSPI data register (SPDR) with the RSPI transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmit buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

#### (2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit, the RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the SPDR register. It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLAi output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.

### (3) Sequence Control

The transfer format that is employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLAi pin output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

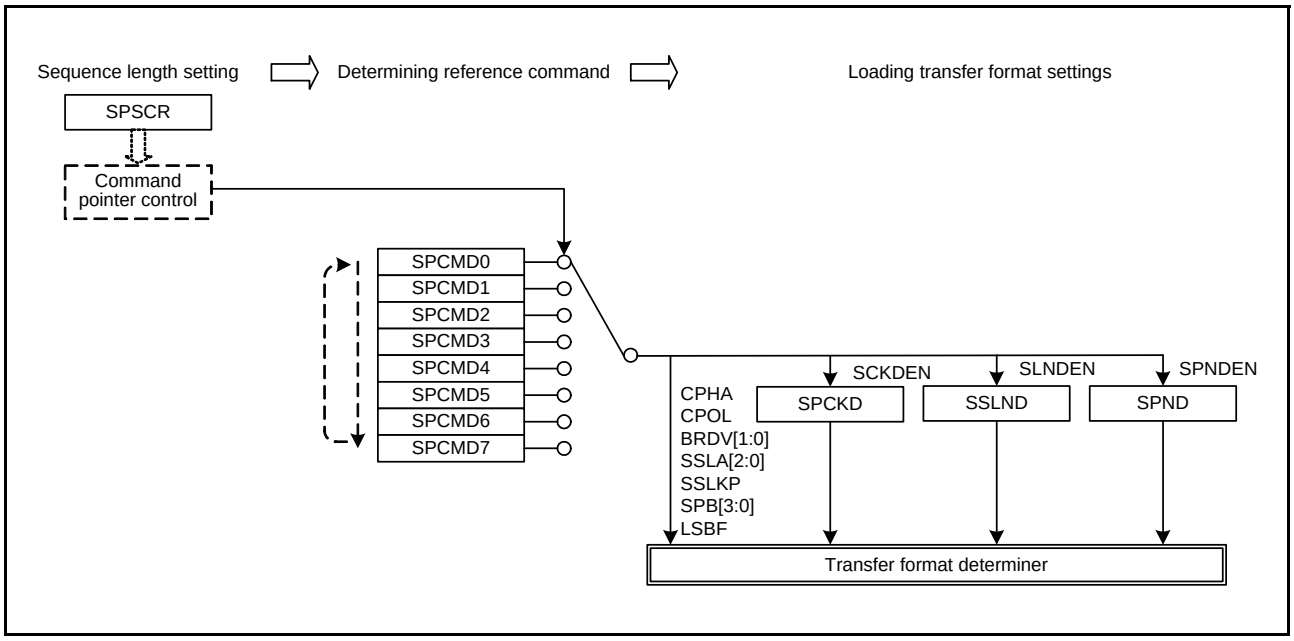



**Figure 38.31 Procedure for Determining the Form of Serial Transfer in Master Mode**

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

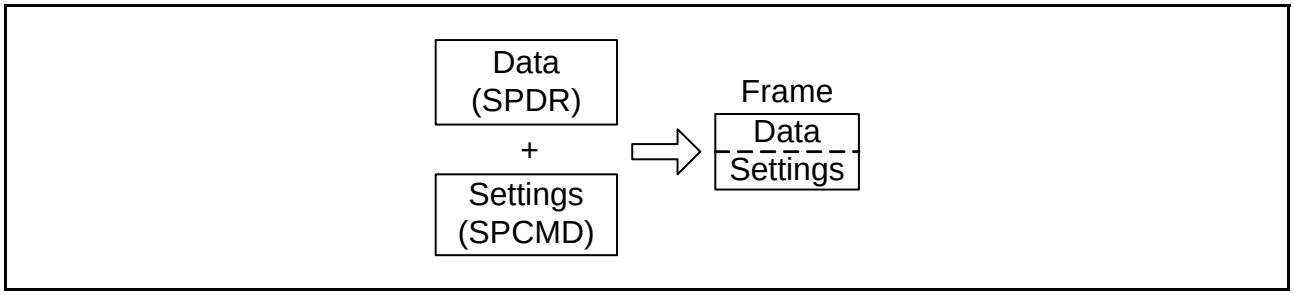



**Figure 38.32 Concept of a Frame**

Figure 38.33 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 38.4.

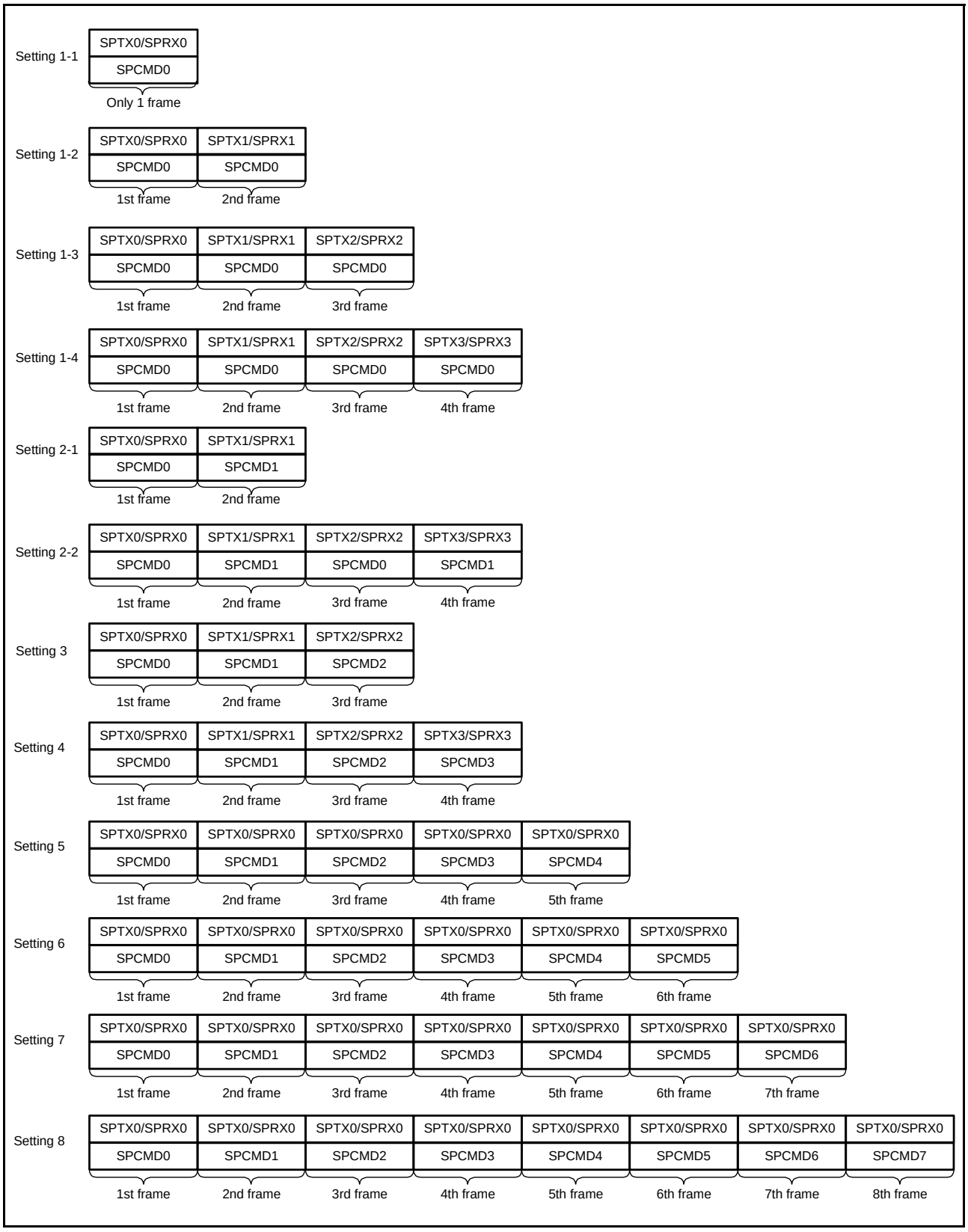



**Figure 38.33 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations**

### (4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).
Figure 38.34 shows an example of an SSLAi signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 38.34. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

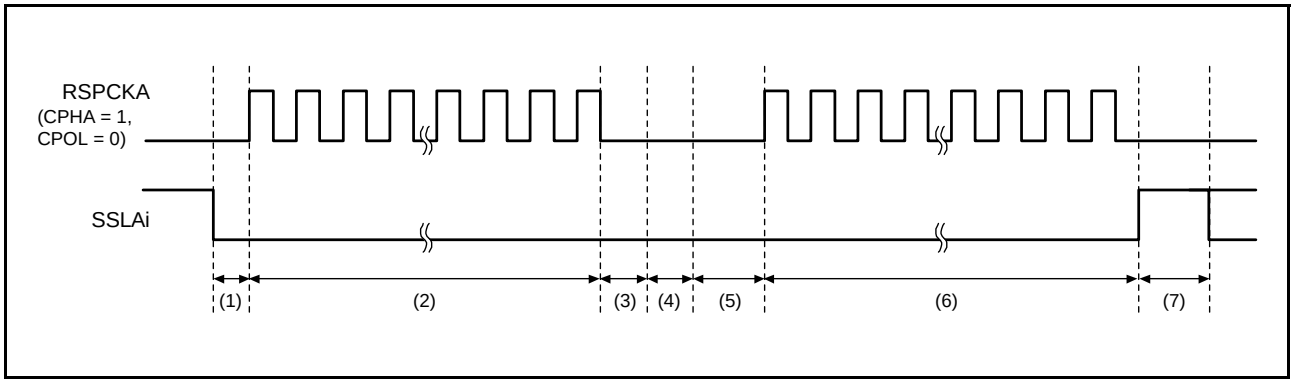



**Figure 38.34 Example of Burst Transfer Operation Using SSLKP Bit**

(1) Based on SPCMD0, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(2) The RSPI executes serial transfers according to SPCMD0.
(3) The RSPI inserts SSL negation delays.
(4) Since the SPCMD0.SSLKP bit is 1, the RSPI keeps the SSLAi signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
(5) Based on SPCMD1, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(6) The RSPI executes serial transfers according to SPCMD1.
(7) Because the SPCMD1.SSLKP bit is 0, the RSPI negates the SSLAi signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLAi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLAi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLAi signal status to SSLAi signal assertion ((5) in Figure 38.34) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLAi signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLAi signal assertion for the next transfer that is detected internally.

### (5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SPCMDm.SCKDEN bit and SPCKD, as listed in Table 38.9. For a definition of RSPCK delay, refer to section 38.3.5, Transfer Format.

**Table 38.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value**

| SPCMDm.SCKDEN Bit | SPCKD.SCKDL[2:0] Bits | RSPCK Delay Value |
|---|---|---|
| 0 | 000b to 111b | 1 RSPCK |
| 1 | 000b | 1 RSPCK |
| | 001b | 2 RSPCK |
| | 010b | 3 RSPCK |
| | 011b | 4 RSPCK |
| | 100b | 5 RSPCK |
| | 101b | 6 RSPCK |
| | 110b | 7 RSPCK |
| | 111b | 8 RSPCK |

### (6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLNDEN bit and SSLND, as listed in Table 38.10. For a definition of SSL negation delay, refer to section 38.3.5, Transfer Format.

**Table 38.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value**

| SPCMDm.SLNDEN Bit | SSLND.SLNDL[2:0] Bits | SSL Negation Delay Value |
|---|---|---|
| 0 | 000b to 111b | 1 RSPCK |
| 1 | 000b | 1 RSPCK |
| | 001b | 2 RSPCK |
| | 010b | 3 RSPCK |
| | 011b | 4 RSPCK |
| | 100b | 5 RSPCK |
| | 101b | 6 RSPCK |
| | 110b | 7 RSPCK |
| | 111b | 8 RSPCK |

### (7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and SPND, as listed in Table 38.11. For a definition of next-access delay, refer to section 38.3.5, Transfer Format.

**Table 38.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value**

| SPCMDm.SPNDEN Bit | SPND.SPNDL[2:0] Bits | Next-Access Delay Value |
|---|---|---|
| 0 | 000b to 111b | 1 RSPCK + 2 PCLK |
| 1 | 000b | 1 RSPCK + 2 PCLK |
| | 001b | 2 RSPCK + 2 PCLK |
| | 010b | 3 RSPCK + 2 PCLK |
| | 011b | 4 RSPCK + 2 PCLK |
| | 100b | 5 RSPCK + 2 PCLK |
| | 101b | 6 RSPCK + 2 PCLK |
| | 110b | 7 RSPCK + 2 PCLK |
| | 111b | 8 RSPCK + 2 PCLK |

### (8) Initialization Flowchart

Figure 38.35 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

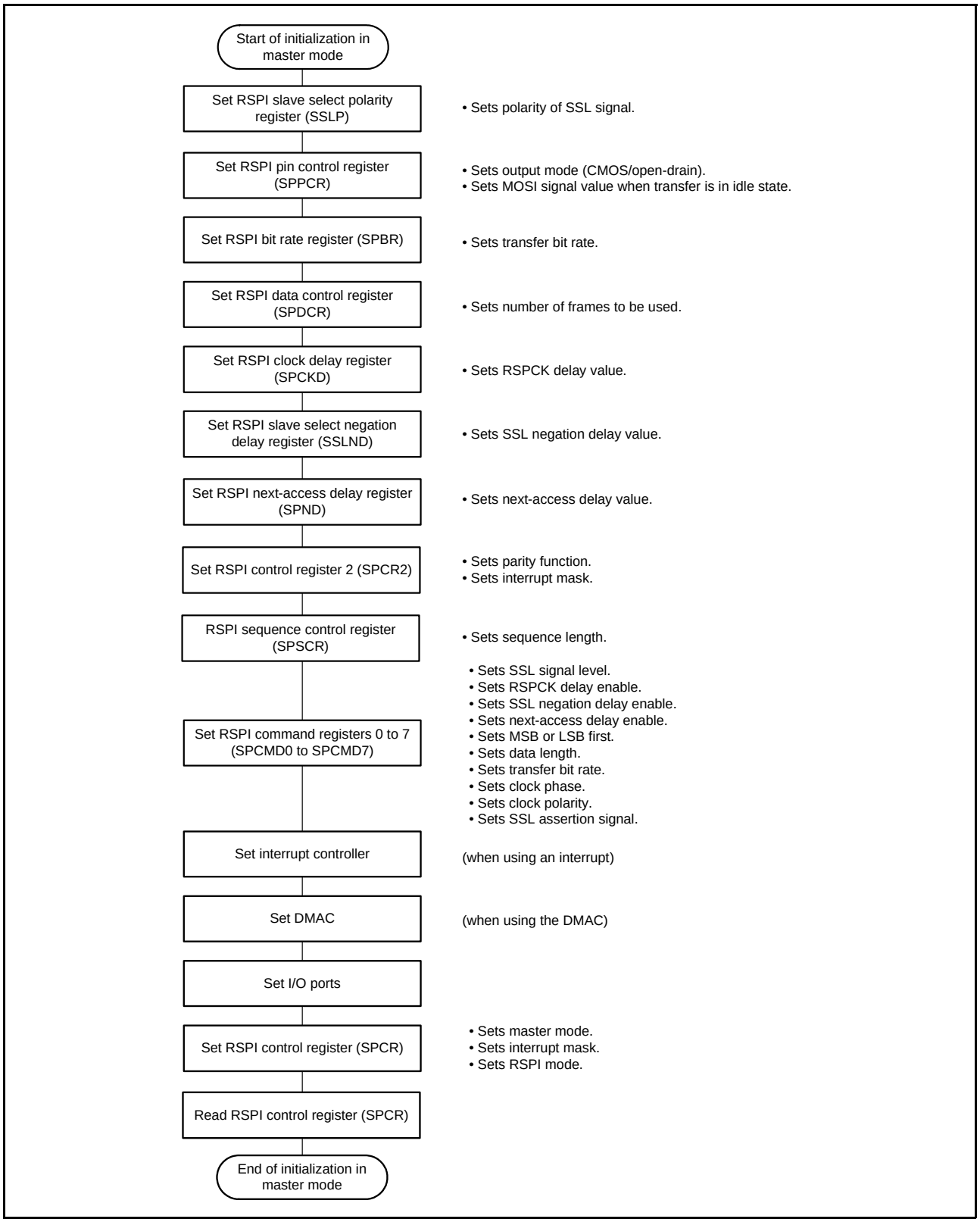


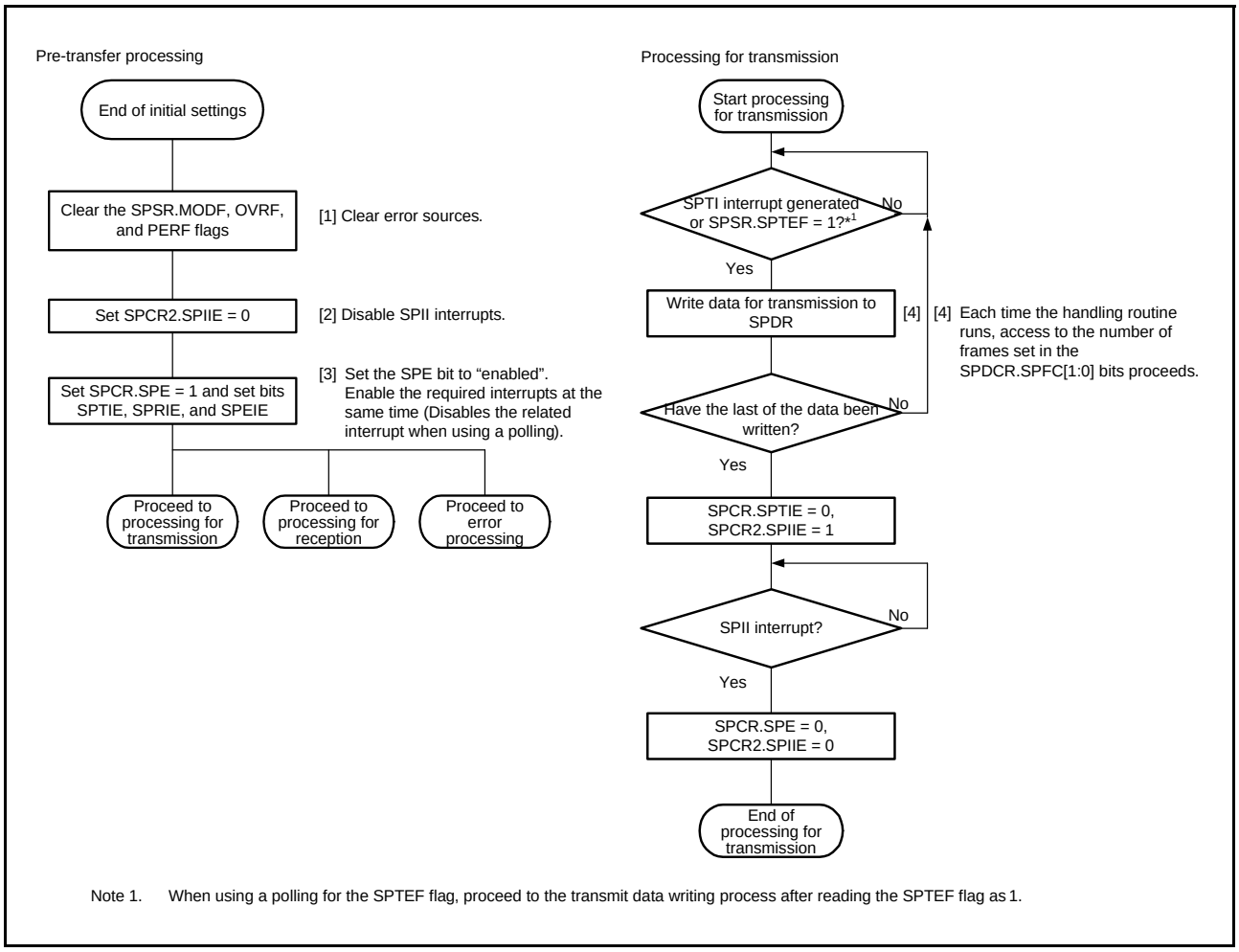

**Figure 38.35 Example of Initialization Flowchart in Master Mode (SPI Operation)**

### (9) Software Processing Flow

Figure 38.36 to Figure 38.38 show examples of the flow of software processing.

### (a) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission by enabling the SPI interrupt after the last writing of data for transmission.
The completion of data transmission can also be checked by polling to see if the SPSR.IDLNF flag has become 0, instead of using the SPII interrupt. However, one cycle of PCLK is required for the time from when data for transmission is written in the SPDR register to when the IDLNF flag becomes 1. After the last data is written in the SPDR register, discard the value of the SPSR register once not to judge the condition with the IDLNF flag which has not yet become 1, and read and use the value of the SPSR.IDLNF flag to confirm the completion of data transmission.




Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

**Figure 38.36 Flowchart in Master Mode (Transmission)**

## (b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

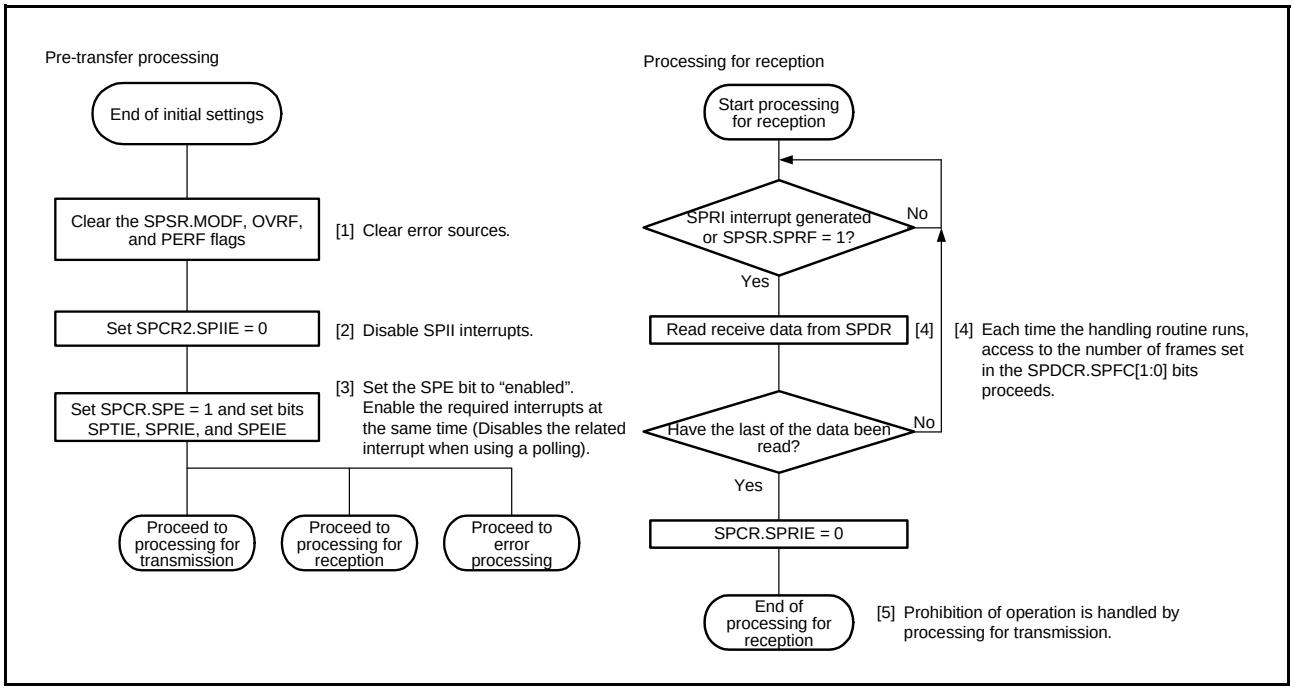



**Figure 38.37 Flowchart in Master Mode (Reception)**

### (c) Flow of Error Processing

The RSPI has three types of error. When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

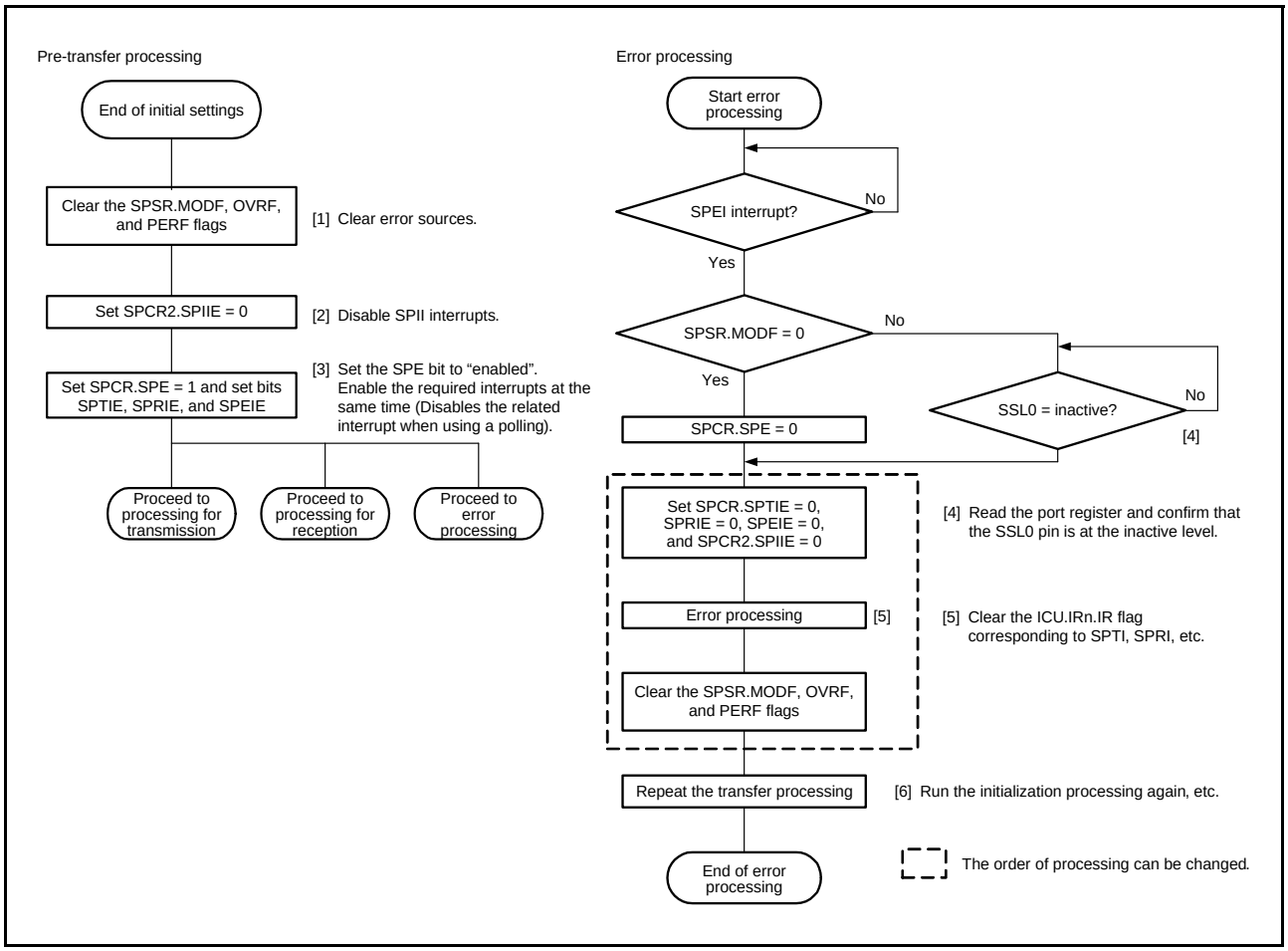



**Figure 38.38 Flowchart for Master Mode (Error Processing)**

### 38.3.10.2 Slave Mode Operation

#### (1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLA0 input signal assertion, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLA0 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCKA edge in an SSLA0 signal asserted condition, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 1, the first RSPCKA edge in an SSLA0 signal asserted condition triggers the start of a serial transfer.
When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register unchanged, in the full state.
Irrespective of the CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSLP.SSL0P bit.

#### (2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 38.3.8, Error Detection).
The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting. The polarity of the SSLA0 input signal depends on the SSLP.SSL0P bit setting.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.

#### (3) Notes on Single-Slave Operations

If the SPCMD0.CPHA bit is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSLA0 input signal. In the type of configuration shown in Figure 38.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLA0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLA0 input signal should not be fixed.

### (4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.
If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

### (5) Initialization Flowchart

Figure 38.39 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

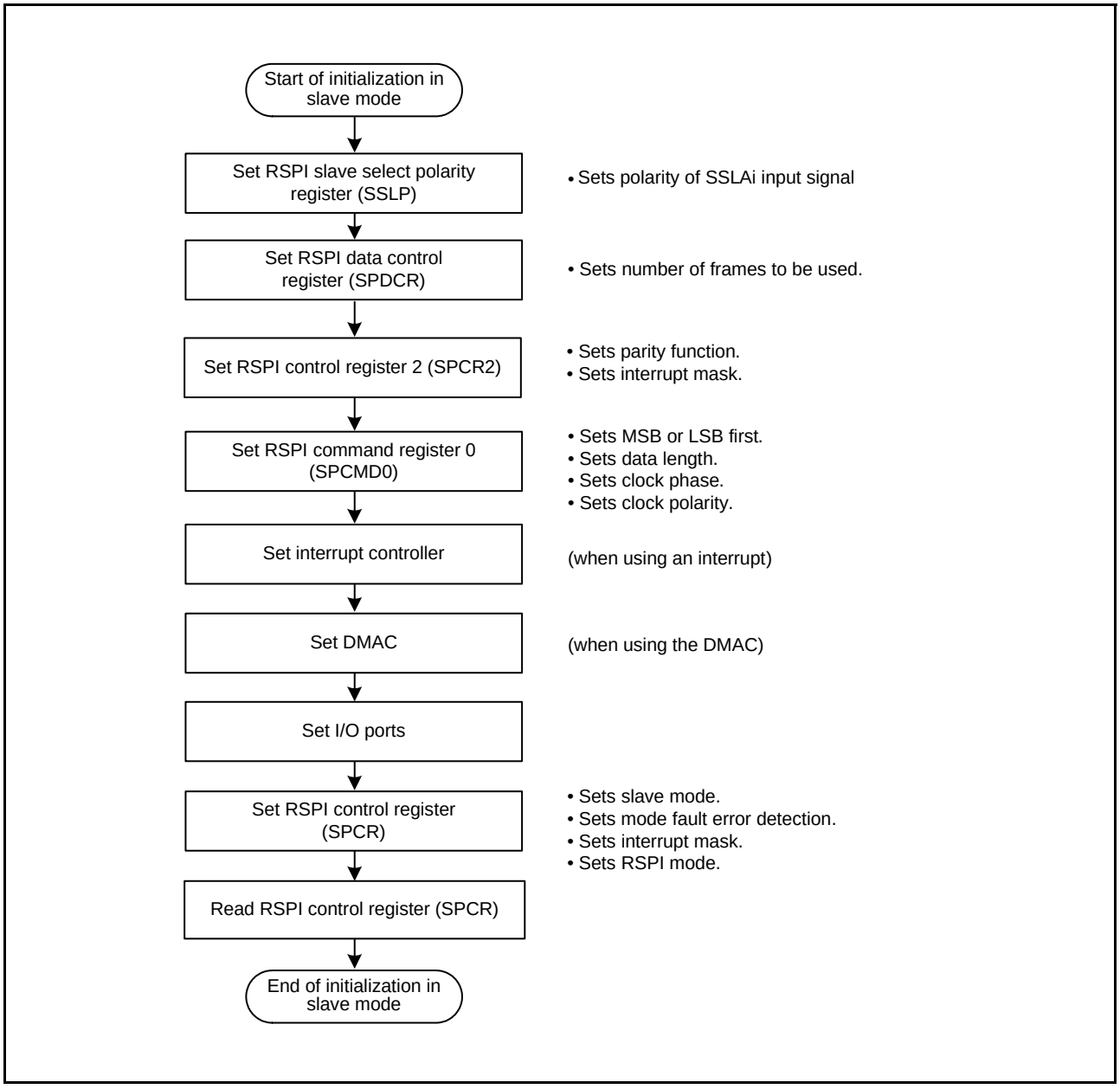



**Figure 38.39 Example of Initialization Flowchart in Slave Mode (SPI Operation)**

### (6) Software Processing Flow

Figure 38.40 to Figure 38.42 show examples of the flow of software processing.

#### (a) Transmit Processing Flow

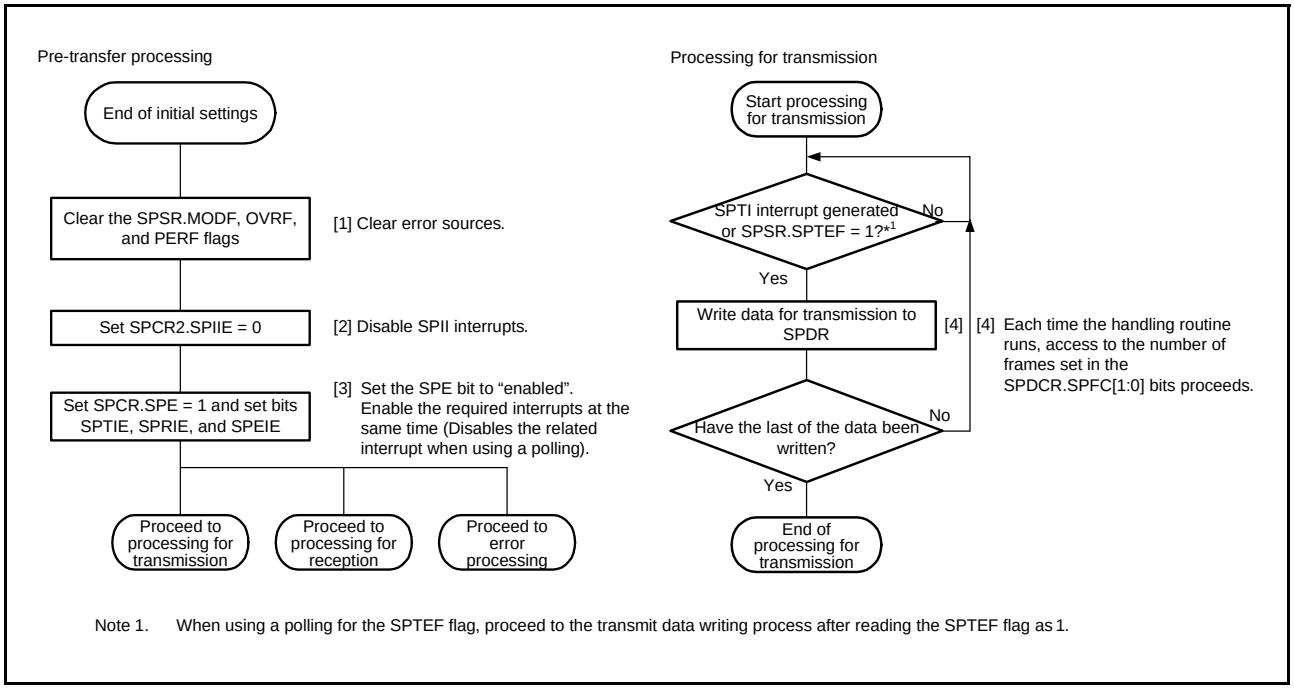



**Figure 38.40 Flowchart in Slave Mode (Transmission)**

#### (b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

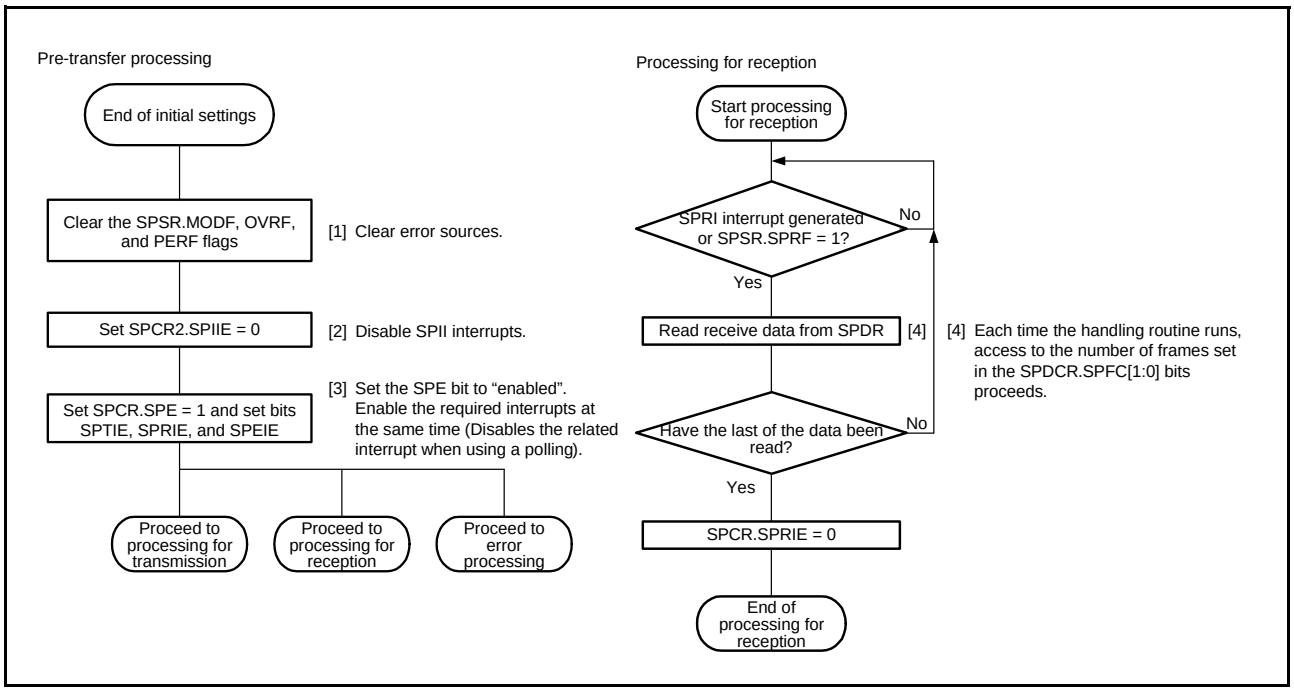



**Figure 38.41 Flowchart in Slave Mode (Reception)**

### (c) Flow of Error Processing

In slave operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the status of the SSLA0 pin.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

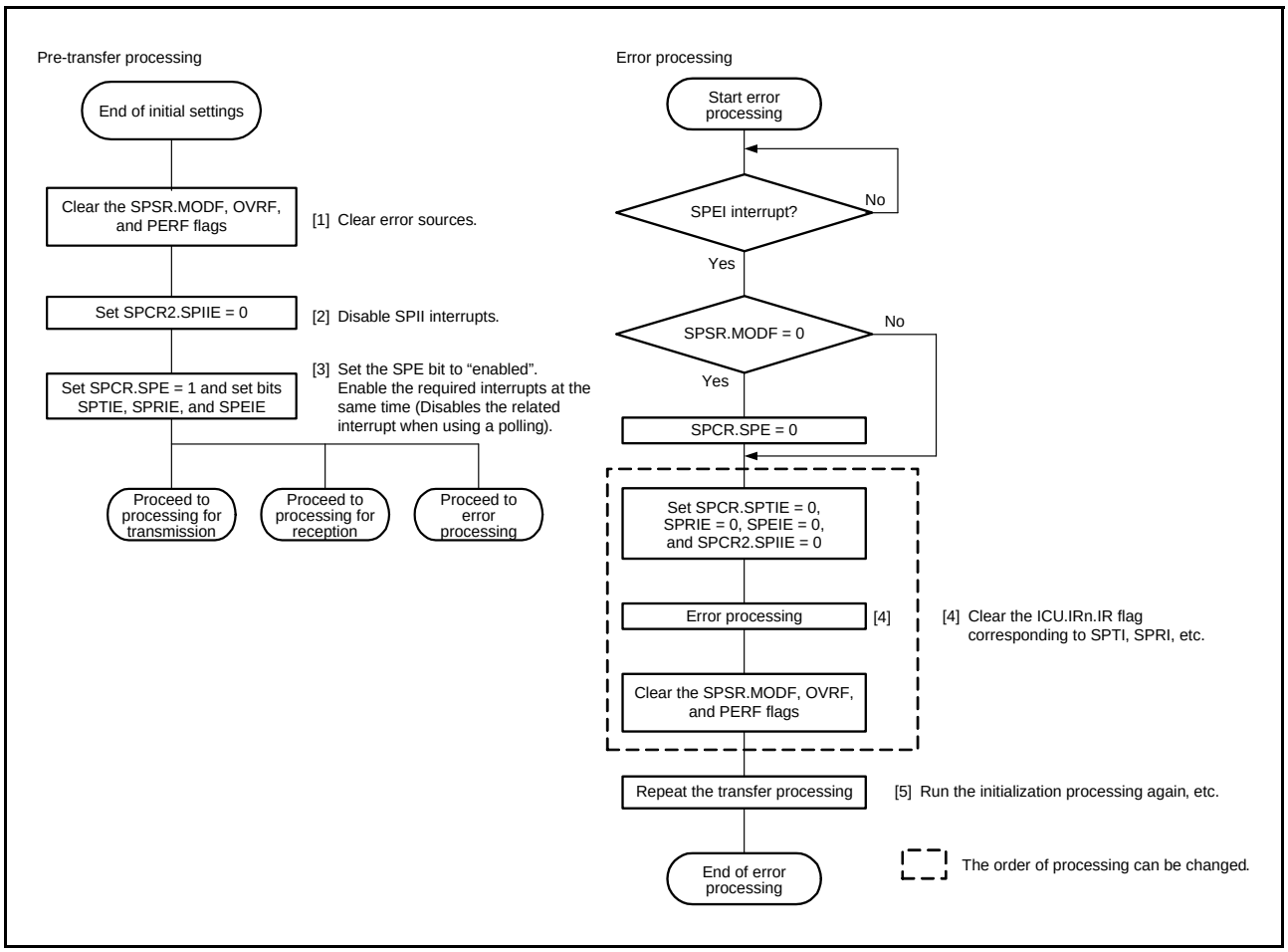



**Figure 38.42 Flowchart for Slave Mode (Error Processing)**

### 38.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLAi pin, operation of the module is the same as in SPI operation. That is, in both master and slave operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLAi pin is not used.
Furthermore, do not set the SPCMDm.CPHA bit to 0 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

#### 38.3.11.1 Master Mode Operation

##### (1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of SPDR when data is written to the SPDR register with the transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

##### (2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

##### (3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLAi signals are not output in clock synchronous operation, these settings are valid.
SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLAi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKA polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.
According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0 register, and in this manner the sequence is executed repeatedly.

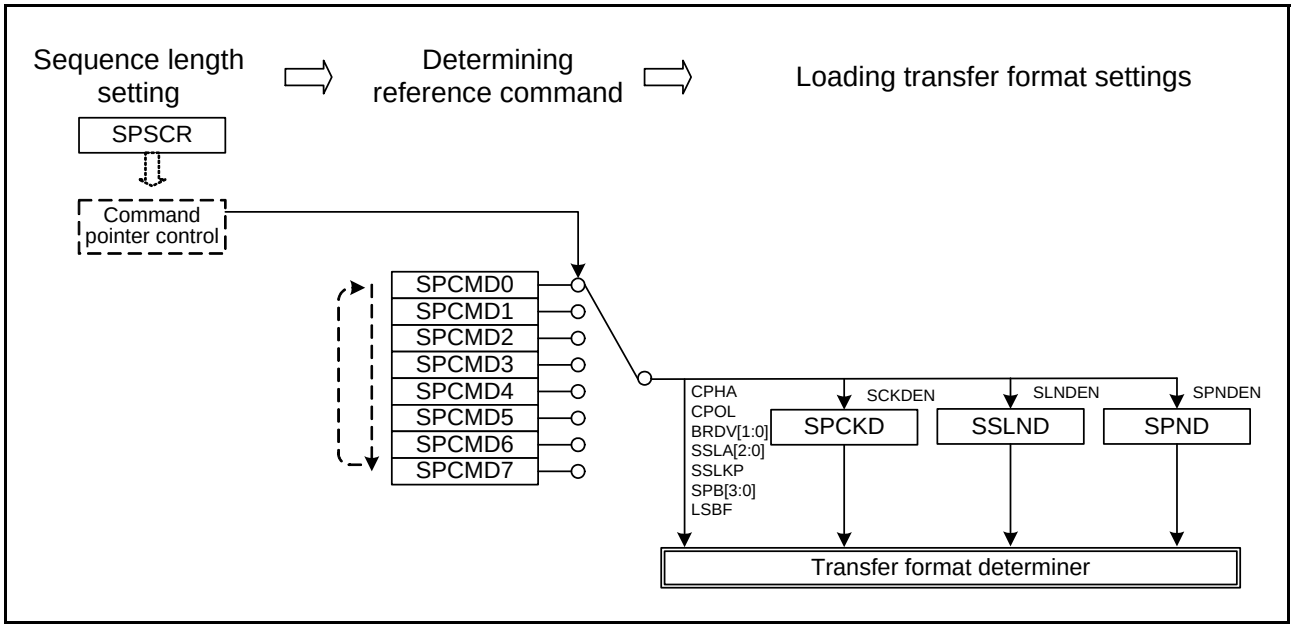



**Figure 38.43 Procedure for Determining the Form of Serial Transmission in Master Mode**

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

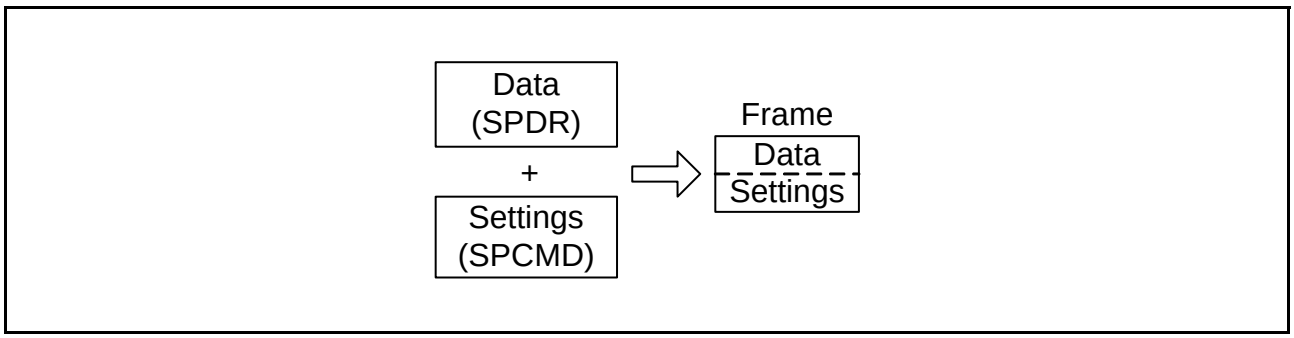



**Figure 38.44 Concept of a Frame**

Figure 38.45 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 38.4.

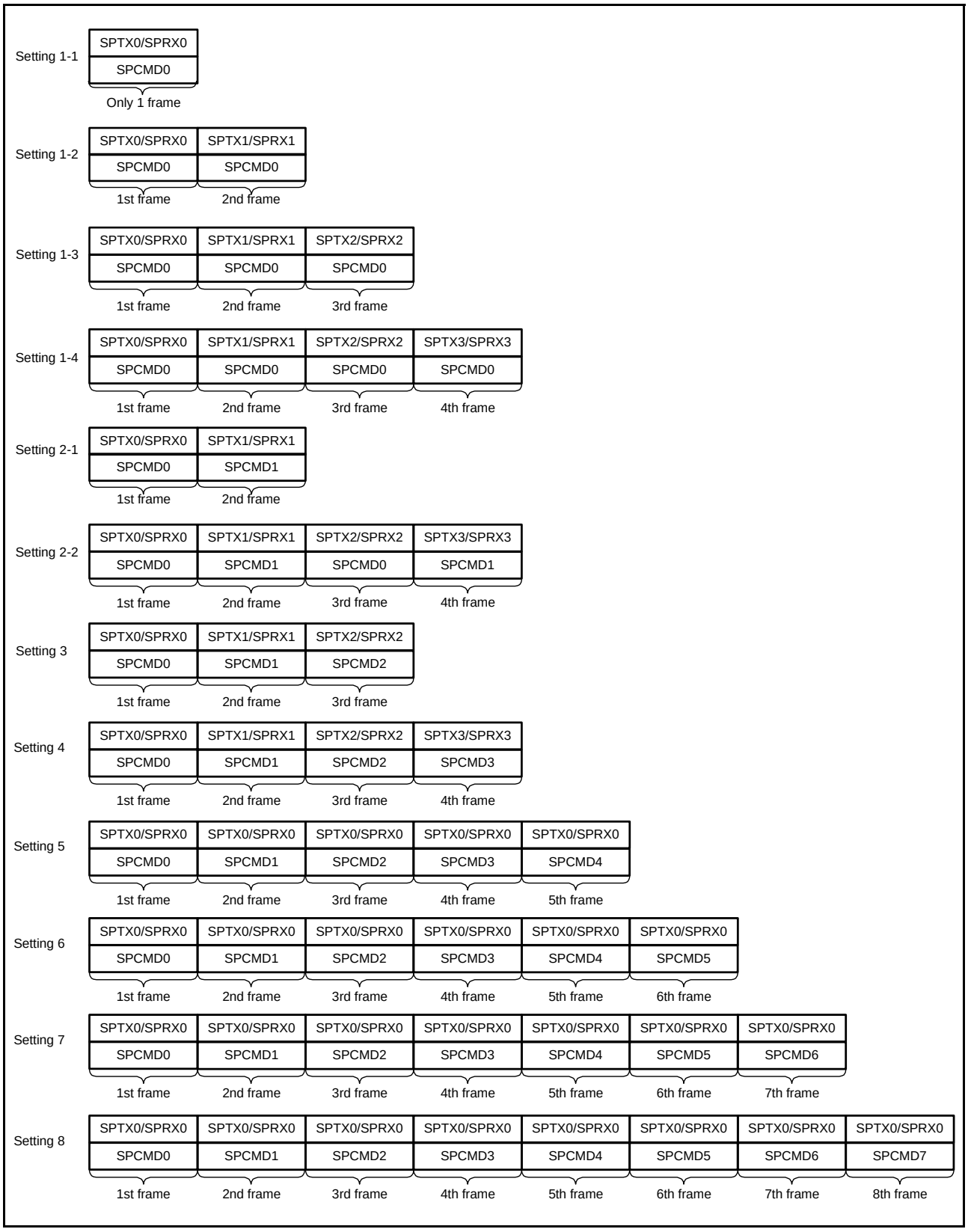



**Figure 38.45 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations**

### (4) Initialization Flowchart

Figure 38.46 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

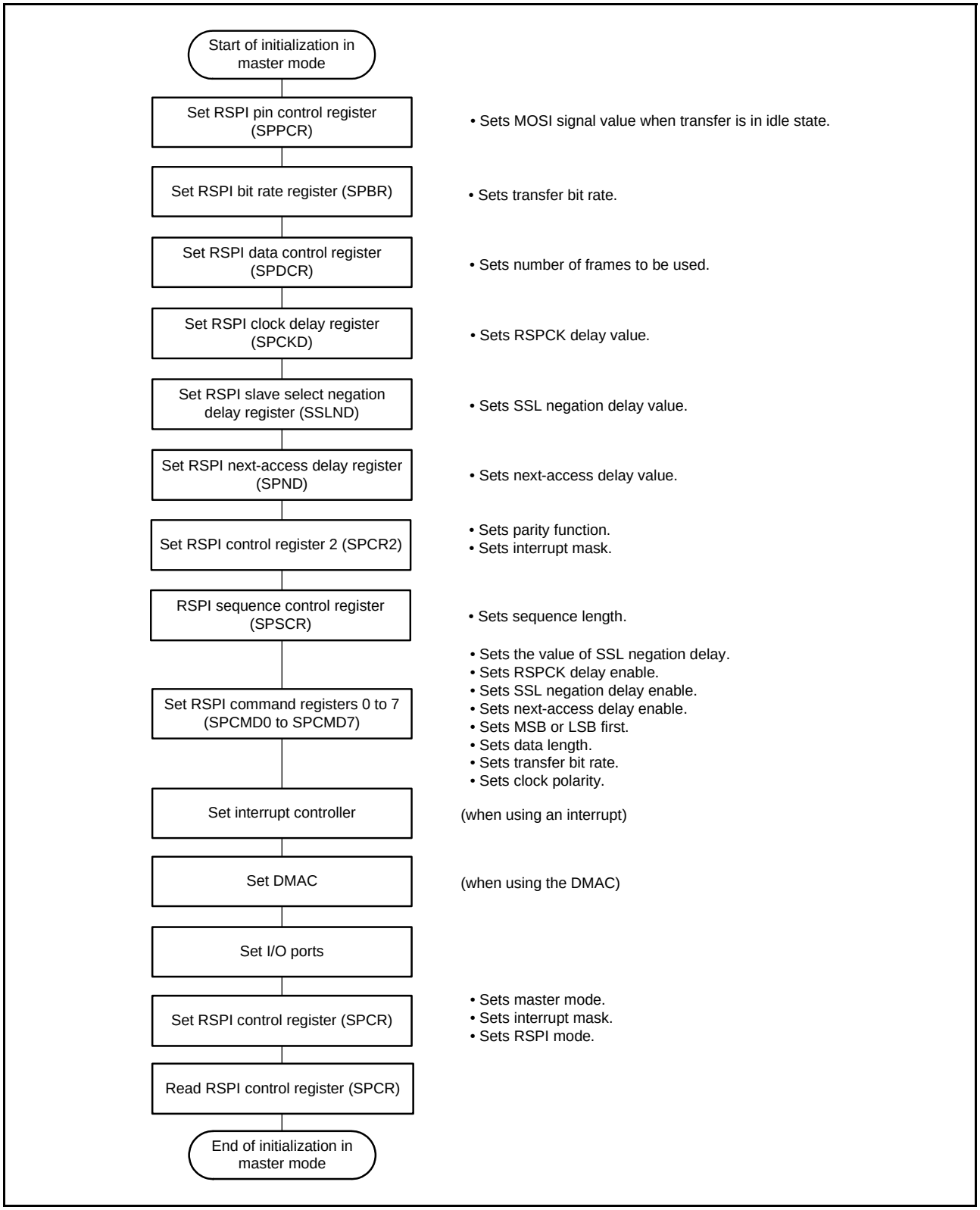



**Figure 38.46 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)**

#### (5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, refer to section 38.3.10.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

### 38.3.11.2 Slave Mode Operation

#### (1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA edge triggers the start of a serial transfer in the RSPI.
When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI keeps the status of the shift register unchanged, in the full state.
When the SPMS bit is 1, the RSPI drives the MISOA output signal.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.
It should be noted that the SSLA0 input signal is not used in clock synchronous operation.

#### (2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format.

### (3) Initialization Flowchart

Figure 38.47 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

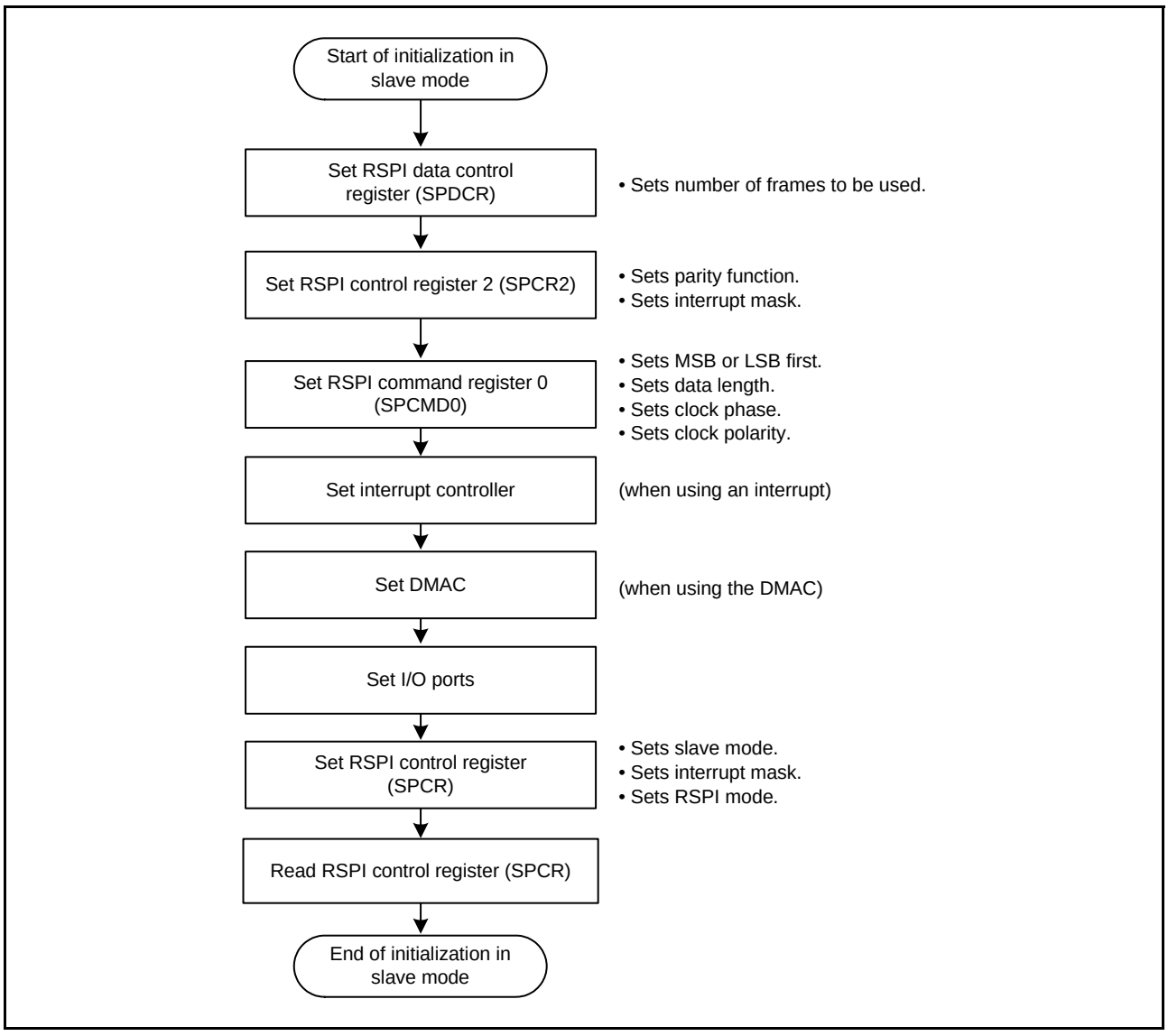



**Figure 38.47 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)**

### (4) Flow of Software Processing

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, refer to section 38.3.10.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

### 38.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSIA pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOA pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI or the reversed transmit data becomes the received data for the RSPI.
Table 38.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 38.48 shows the configuration of the shift register I/O paths for the case where the RSPI in master mode is set in loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

**Table 38.12 SPLP2 and SPLP Bit Settings and Received Data**

| SPPCR.SPLP2 Bit | SPPCR.SPLP Bit | Received Data |
|---|---|---|
| 0 | 0 | Input data from the MOSIA pin or MISOA pin |
| 0 | 1 | Inverted transmit data |
| 1 | 0 | Transmit data |
| 1 | 1 | Transmit data |

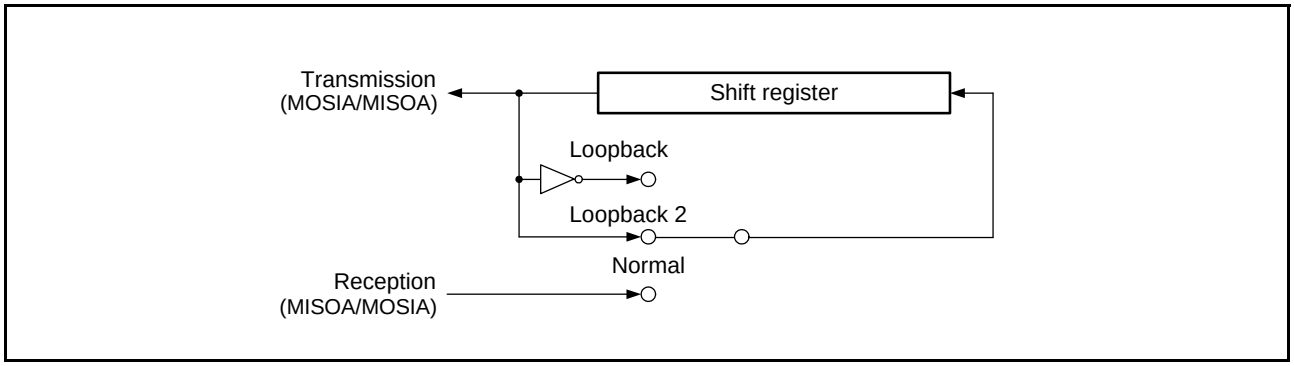



**Figure 38.48 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)**

### 38.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 38.49.

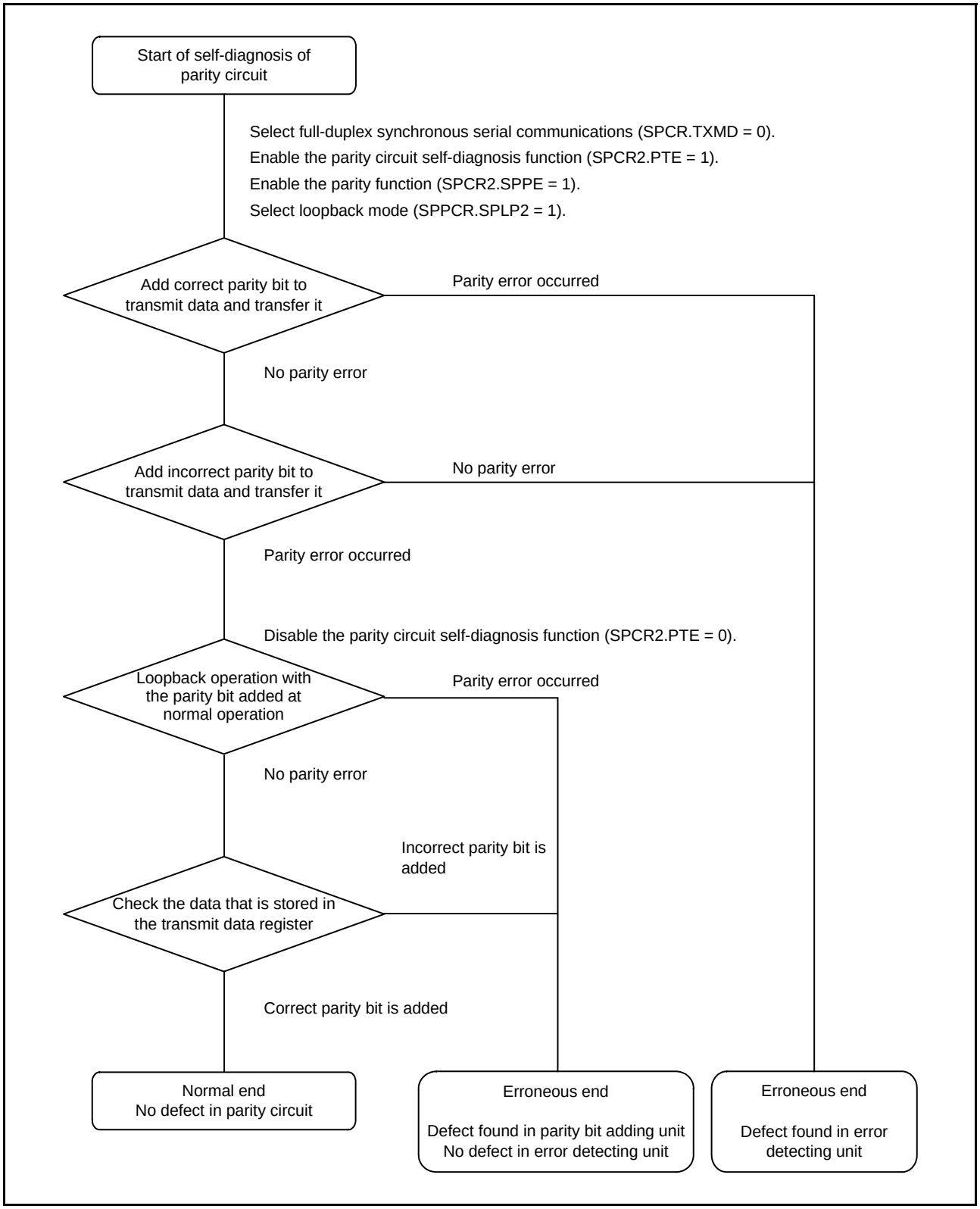



**Figure 38.49 Flowchart for Self-Diagnosis of Parity Circuit**

### 38.3.14 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, mode fault, overrun, parity error, and RSPI idle. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode fault, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 38.13. An interrupt is generated on satisfaction of an interrupt condition in Table 38.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission/reception, the DTC or DMAC must be set up first to be in a status in which transfer is enabled before making the RSPI settings. For the method for setting the DTC or DMAC, refer to section 18, DMA Controller (DMACA), or section 19, Data Transfer Controller (DTCa).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

**Table 38.13 Interrupt Sources of RSPI**

| Interrupt Source | Symbol | Interrupt Condition | DMAC/DTC Activation |
|---|---|---|---|
| Receive buffer full | SPRI | The receive buffer becomes full (the SPRF flag becomes 1) while the SPCR.SPRIE bit is 1. | Possible |
| Transmit buffer empty | SPTI | The transmit buffer becomes empty (the SPTEF flag becomes 1) while the SPCR.SPTIE bit is 1. | Possible |
| RSPI errors (mode fault, overrun and parity error) | SPEI | The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1. | Impossible |
| RSPI idle | SPII | The SPSR.IDLNF flag is set to 0 while the SPCR2.SPIIE bit is 1. | Impossible |

## 38.4 Link Operation by Event Linking

The RSPI0 supports the following event output for the event link controller (ELC). The event link output signal is output regardless of the interrupt enable bit setting.

### 38.4.1 Receive Buffer Full Event Output

This event signal is output when received data have been transferred from the shift register to the SPDR on completion of serial transfer.

### 38.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission have been transferred from the transmit buffer to the shift register and when the value of the SPE bit has changed from 0 to 1.

### 38.4.3 Mode Fault, Overrun, or Parity Error Event Output

#### (1) Mode Fault

Table 38.14 lists the occurrence conditions of a mode fault event.

**Table 38.14 Occurrence Conditions of Mode Fault Event**

| | SPCR.MODFEN Bit | SSLA0 Pin | Remarks |
|---|---|---|---|
| Master (SPCR.MSTR bit = 1) | 1 | Active | Under the condition (the SPCR.MSTR bit is 1 and the MODFEN bit is 1), if the SPCR.SPMS bit is 0, mode fault error, overrun error, and parity error event output cannot be used.<br>Do not set the ELSRn register to 52h. |
| Slave (SPCR.MSTR bit = 0) | 1 | Not active | Event is output only when the pin is deactivated during transmission. |

#### (2) Overrun

The condition for this event signal being output in response to an overrun is completion of serial transfer while the receive buffer contains data that have not been read and the value of the SPCR.TXMD bit is 0, in which case the OVRF flag is set to 1.

#### (3) Parity Error

The condition for this event signal being output in response to a parity error is detection of a parity error on completion of serial transfer while the value of the TXMD bit in SPCR is 0 and the value of the SPPE bit in SPCR2 is 1.

### 38.4.4 RSPI Idle Event Output

#### (1) In Master Mode

In master mode, an event is output when the condition for setting the IDLNF flag (RSPI idle flag) to 0 is satisfied.

#### (2) In Slave Mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (RSPI is initialized).

### 38.4.5 Transmission-Completed Event Output

During both SPI operation and clock synchronous operation in master mode, an event is output under the condition for setting the IDLNF flag (RSPI idle flag) from 1 to 0.

**Table 38.15 Conditions for Generation of a Transmission-Completed Event (Slave)**

| | Transmit Buffer State | Shift Register State | Others |
|---|---|---|---|
| SPI operation (SPMS = 0) | Empty | Empty | Negation of SSLA0 input |
| Clock synchronous operation (SPMS = 1) | Empty | Empty | Edge detection of the last RSPCKA |

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit in transmission or the SPCR.SPE bit is cleared by the mode fault error.

## 38.5 Usage Notes

### 38.5.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can be used to enable or disable the RSPI. Immediately after a reset, operation of the RSPI is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

### 38.5.2 Note on Low Power Consumption Functions

When using the module stop function and entering a low power consumption mode other than sleep mode, set the SPCR.SPE bit to 0 before completing communication.

### 38.5.3 Notes on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started, an interrupt request is internally retained after transfer starts, and this can lead to unanticipated behavior of the ICU.IRn.IR flag.
When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

### 38.5.4 Notes on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

# 39. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

## 39.1 Overview

Table 39.1 lists the specifications of the CRC calculator, and Figure 39.1 shows a block diagram of the CRC calculator.

**Table 39.1 CRC Specifications**

| Item | Description |
|---|---|
| Data for CRC calculation*1 | CRC codes are generated for any desired data in 8n-bit units (where n is a whole number) |
| CRC processor unit | 8-bit parallel processing |
| CRC generating polynomial | One of three generating polynomials is selectable<br>• 8-bit CRC<br>$X^8 + X^2 + X + 1$<br>• 16-bit CRC<br>$X^{16} + X^{15} + X^2 + 1$<br>$X^{16} + X^{12} + X^5 + 1$ |
| CRC calculation switching | The bit order of CRC calculation results can be switched for LSB first or MSB first communication |
| Low power consumption function | Module stop state can be set. |

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit units.

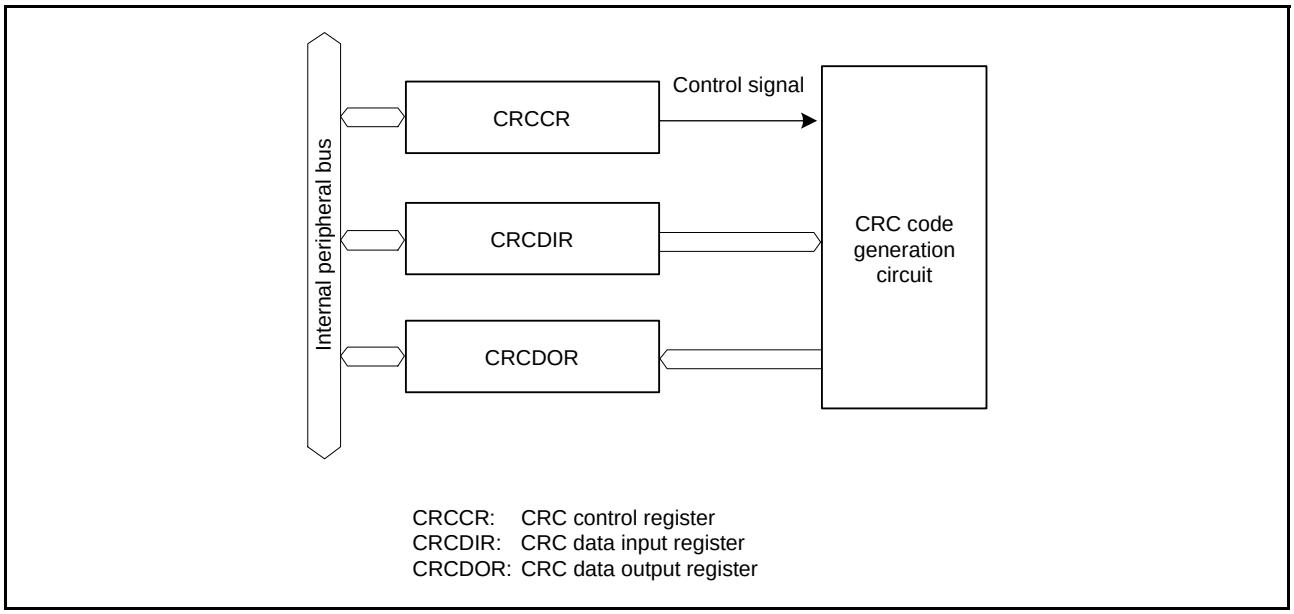



**Figure 39.1 CRC Block Diagram**

## 39.2 Register Descriptions

### 39.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DORCLR | — | — | — | — | LMS | GPS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | GPS[1:0] | CRC Generating Polynomial Switching | b1 b0<br>0 0: No calculation is executed.<br>0 1: 8-bit CRC ($X^8 + X^2 + X + 1$)<br>1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$)<br>1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$) | R/W |
| b2 | LMS | CRC Calculation Switching | 0: Generates CRC for LSB first communication.<br>1: Generates CRC for MSB first communication. | R/W |
| b6 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | DORCLR | CRCDOR Register Clear | 1: Clears the CRCDOR register.<br>This bit is read as 0. | R/W*1 |

Note 1. Only 1 can be written.

**LMS Bit (CRC Calculation Switching)**

This bit selects the bit order of generated 16-bit CRC code. Transmit the lower-order byte (b7 to b0) of the CRC code first for LSB first communication and the higher-order byte (b15 to b8) first for MSB first communication. For details on the transmission and reception of CRC codes, refer to section 39.3, Operation.

**DORCLR Bit (CRCDOR Register Clear)**

Write 1 to this bit so that the CRCDOR register is set to 0000h.
This bit is read as 0. Only 1 can be written.

### 39.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CRCDIR is a readable and writable register. Write data for CRC calculation to this register.

### 39.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CRCDOR is a readable and writable register.
Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.
Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.

## 39.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared before CRC calculation.
When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of the CRCDOR register.

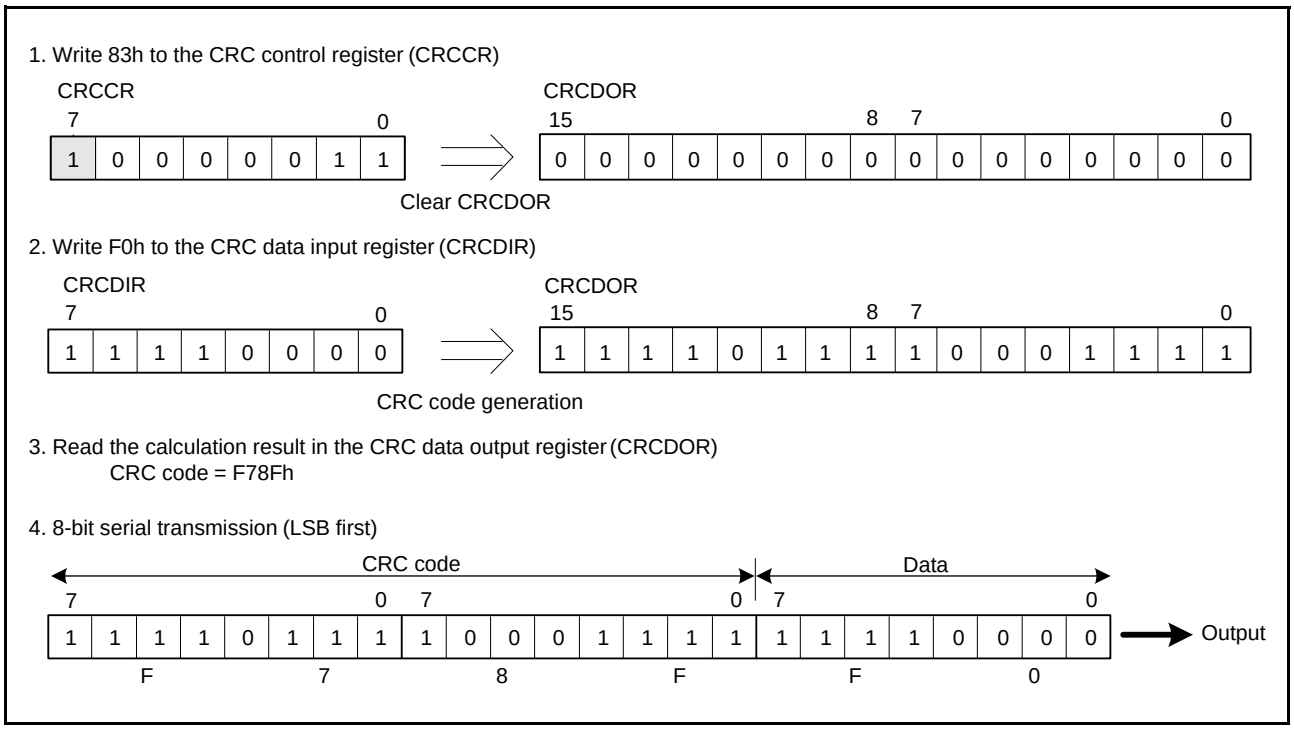



**Figure 39.2 LSB First Data Transmission**

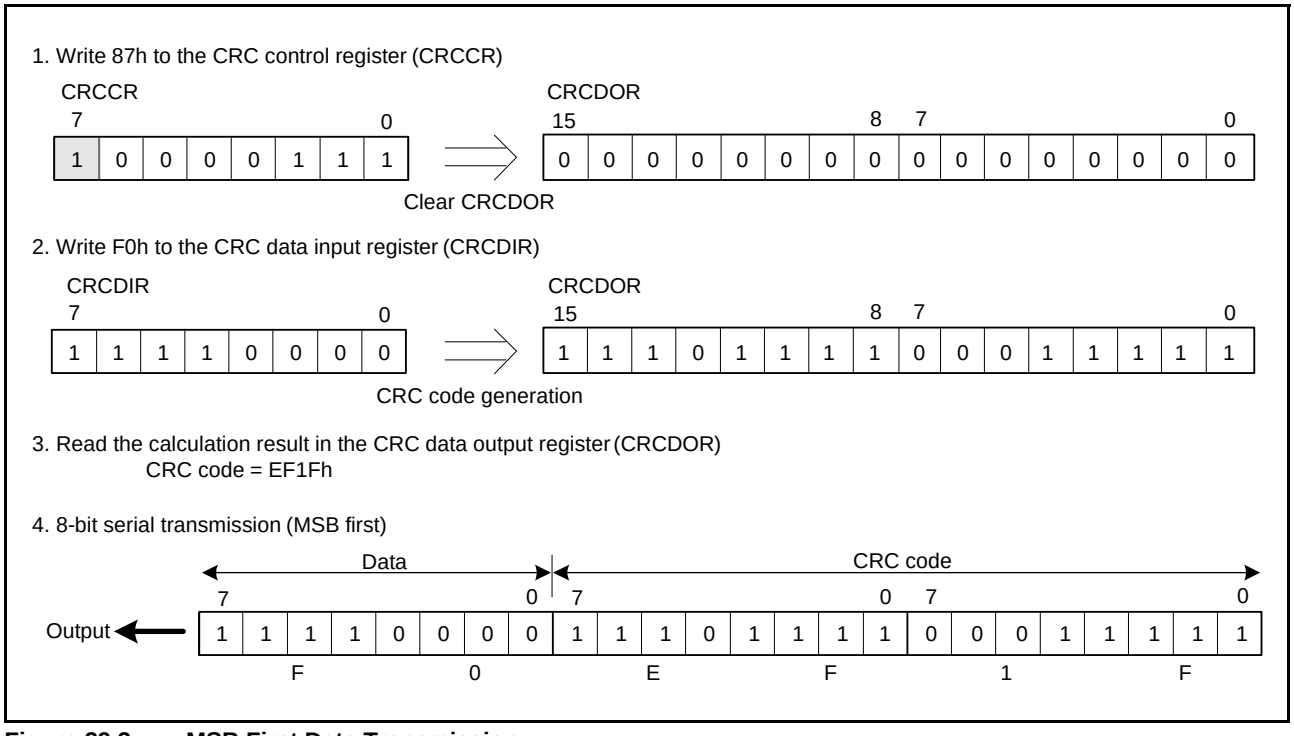



**Figure 39.3 MSB First Data Transmission**

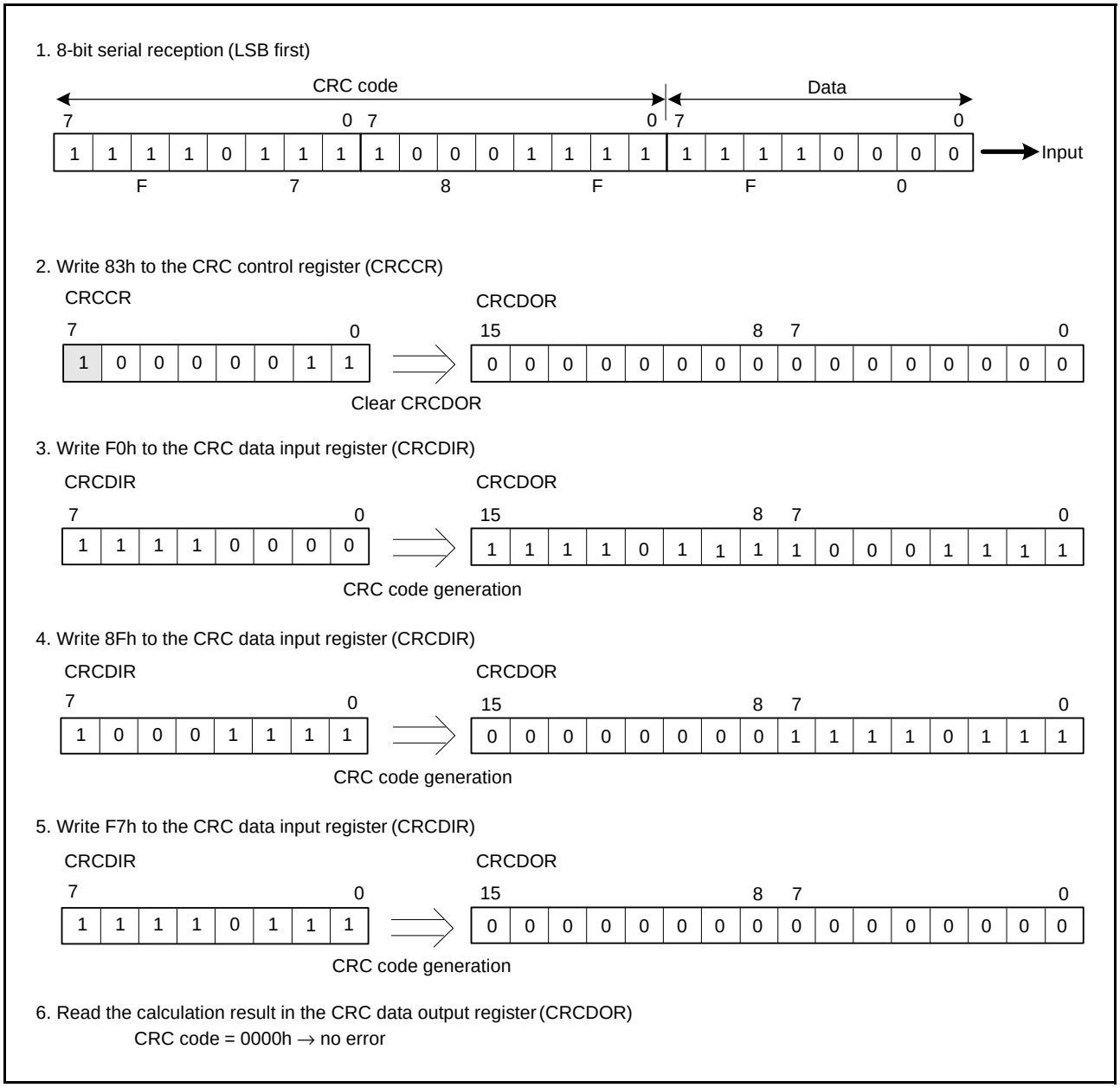



**Figure 39.4 LSB First Data Reception**

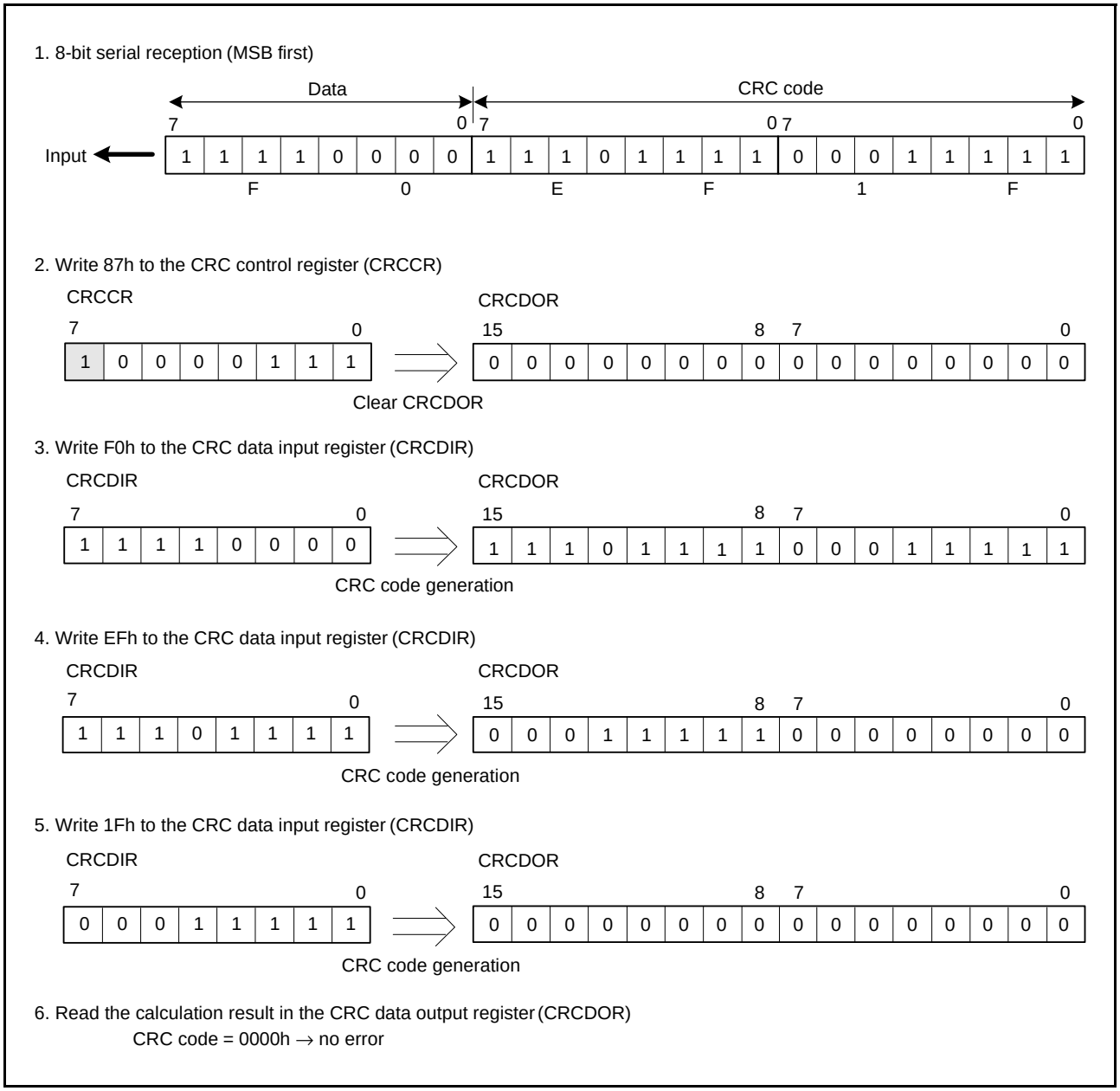



**Figure 39.5 MSB First Data Reception**

## 39.4 Usage Notes

### 39.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

### 39.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

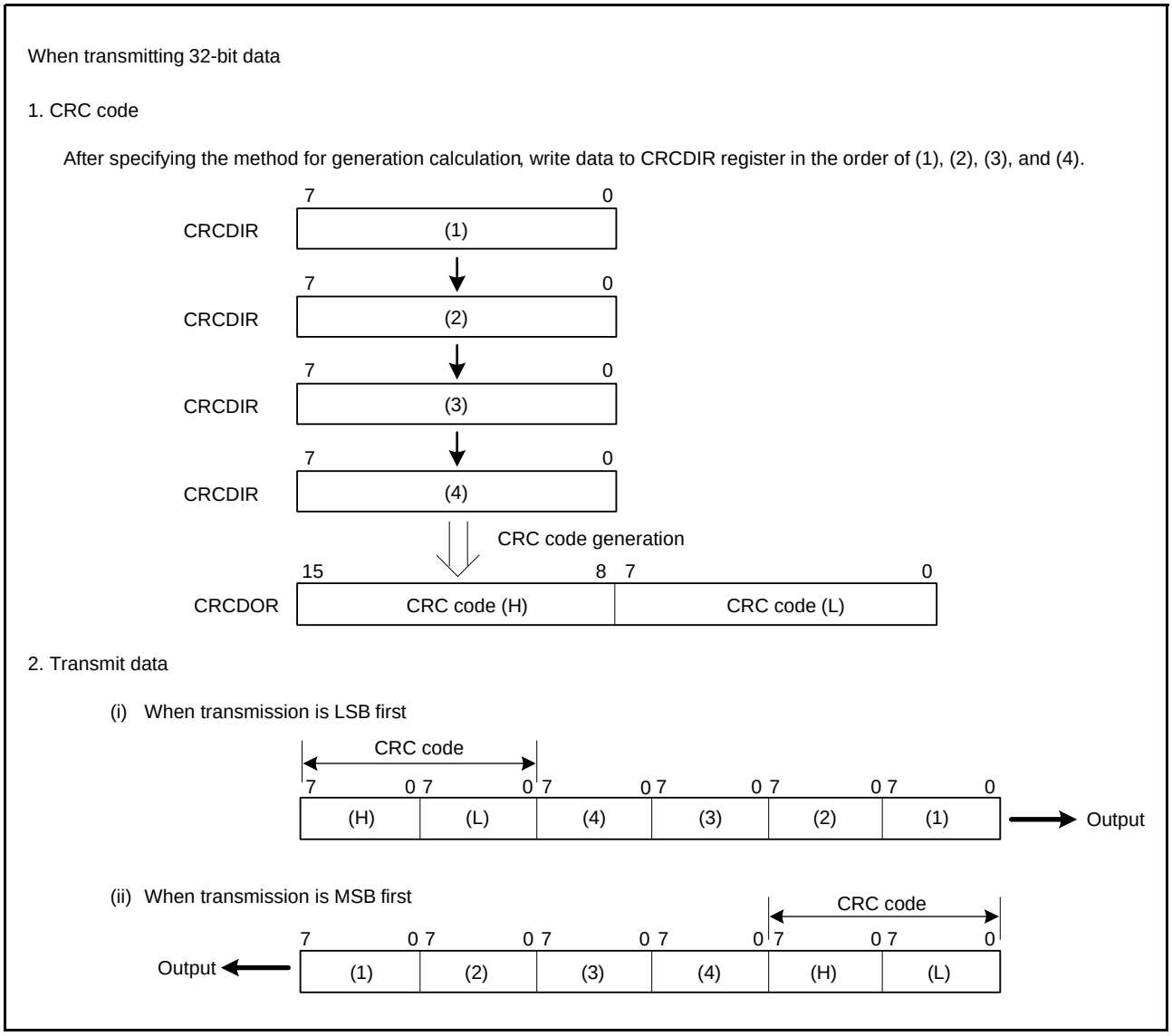



**Figure 39.6 LSB First and MSB First Data Transmission**

# 40. SD Host Interface (SDHIa)

This MCU incorporates an SD host interface (SDHI) which is compliant with the SD Specifications. When developing host devices that are compliant with the SD Specifications, the user must enter into the SD Host/Ancillary Product License Agreement (SD HALA).

## 40.1 Overview

Table 40.1 lists the SDHI specifications.

**Table 40.1 SDHI Specifications**

| Item | Description |
|---|---|
| SD bus interface | • Compatible with SD memory card and SDIO card (**NOT** compatible with the SPI bus interface, embedded SDIO shared bus, 8-bit SD bus, or SDIO suspend/resume functions)<br>• Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode<br>• Compatible with SD, SDHC, and SDXC formats |
| Transfer modes | Supports default speed mode |
| SDHI clock | The SDHI clock is generated by dividing peripheral module clock B (PCLKB) by *n*, where *n* = 1, 2, 4, 8, 16, 32, 64, 128, 256, or 512 |
| Error check functions | • CRC7 (command/response)<br>• CRC16 (transfer data) |
| Interrupt sources | • Card access interrupt (CACI)<br>• SDIO access interrupt (SDACI)<br>• Card detection interrupt (CDETI)<br>• SD buffer access interrupt (SBFAI) |
| DMA transfer sources | • DMAC and DTC triggerable by the SBFAI interrupt<br>• SD buffer is read and write accessible using the DMAC and DTC |
| Other functions | • Card detection<br>• Write protection |

Figure 40.1 shows a block diagram of the SDHI.

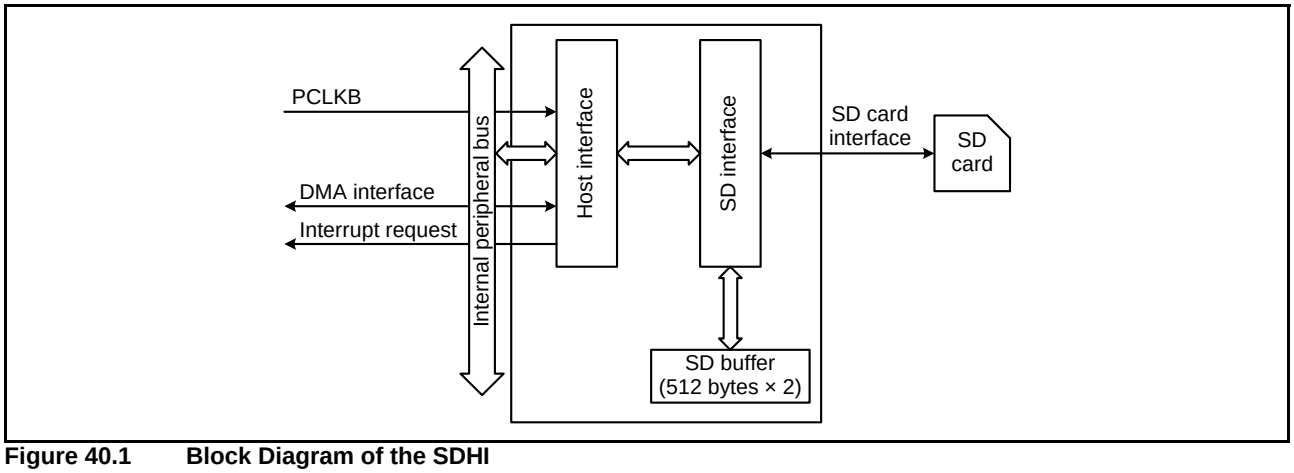



**Figure 40.1 Block Diagram of the SDHI**

Table 40.2 lists the pin configuration of the SDHI.

**Table 40.2 Pin Configuration of the SDHI**

| Pin Name | I/O | Description |
|---|---|---|
| SDHI_CLK | Output | SDHI clock |
| SDHI_CMD | I/O | Command output, response input |
| SDHI_D0 | I/O | Data 0 (DAT0) |
| SDHI_D1 | I/O | Data 1 (DAT1), SDIO access interrupt |
| SDHI_D2 | I/O | Data 2 (DAT2), read wait |
| SDHI_D3 | I/O | Data 3 (DAT3), SD card detection |
| SDHI_CD | Input | SD card detection |
| SDHI_WP | Input | SD card write protection |

## 40.2 Register Details

### 40.2.1 Command Register (SDCMD)

Address(es): SDHI.SDCMD 0008 AC00h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CMD12AT[1:0] | | TRSTP | CMDRW | CMDTP | RSPTP[2:0] | | | ACMD[1:0] | | CMDIDX[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | CMDIDX[5:0] | Command Index Field Value Select | These bits configure the command index field value. The examples below include the bit values for the ACMD[1:0] bits.<br>b7 b0<br>0 0 0 0 0 1 1 0: CMD6<br>0 0 0 1 0 0 1 0: CMD18<br>0 1 0 0 1 1 0 1: ACMD13 | R/W |
| b7, b6 | ACMD[1:0] | Command Type Select | b7 b6<br>0 0: CMD<br>0 1: ACMD<br>Only set the values listed above. | R/W |
| b10 to b8 | RSPTP[2:0] | Response Type Select *1 | b10 b8<br>0 0 0: Normal mode. Depending on the command, the response type and transfer method are selected by setting the ACMD[1:0] bits and CMDIDX[5:0] bits. At this time, the values for b15 to b11 in this register are invalid.<br>0 1 1: Expansion mode and no response<br>1 0 0: Expansion mode and R1, R5, R6, or R7 response<br>1 0 1: Expansion mode and R1b response<br>1 1 0: Expansion mode and R2 response<br>1 1 1: Expansion mode and R3 or R4 response<br>Only set the values listed above. | R/W |
| b11 | CMDTP | Data Transfer Select *2 | 0: Command does not include data transfer (bc, bcr, or ac)<br>1: Command includes data transfer (adtc) | R/W |
| b12 | CMDRW | Data Transfer Direction Select *3 | 0: Write data to the SD card<br>1: Read data from the SD card | R/W |
| b13 | TRSTP | Block Transfer Select *3 | 0: Single block transferred<br>1: Multiple blocks transferred | R/W |
| b15, b14 | CMD12AT[1:0] | CMD12 Automatic Issue Select *4 | b15 b14<br>0 0: CMD12 is automatically issued during multi-block transfer<br>0 1: CMD12 is not automatically issued during multi-block transfer<br>Only set the values listed above. | R/W |
| b31 to b16 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. Some commands cannot be used in normal mode. Refer to Table 40.3 and set the RSPTP[2:0] bits.
Note 2. The CMDTP bit is valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. Bits CMDRW and TRSTP are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SDCMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. Refer to Table 40.3 for setting examples. Do not write to the SDCMD register when the SDSTS2.CBSY flag is 1.

Table 40.3 lists examples of SDCMD register settings.

**Table 40.3 Examples of SDCMD Register Settings**

| Type | Command Symbol | SDCMD Register Setting | Remarks |
|---|---|---|---|
| CMD | CMD0 | 0000 0000h | |
| | CMD2 | 0000 0002h | |
| | CMD3 | 0000 0003h | |
| | CMD4 | 0000 0004h | |
| | CMD5 | 0000 0705h or 0000 0005h | |
| | CMD6 | 0000 1C06h or 0000 0006h | |
| | CMD7 | 0000 0007h | When the card is deselected, the SD card does not return a response, so the SDSTS2.RSPTO flag becomes 1. |
| | CMD8 | 0000 0408h or 0000 0008h | |
| | CMD9 | 0000 0009h | |
| | CMD10 | 0000 000Ah | |
| | CMD11 | 0000 040Bh or 0000 000Bh | |
| | CMD12 | 0000 000Ch | |
| | CMD13 | 0000 000Dh | |
| | CMD15 | 0000 000Fh | |
| | CMD16 | 0000 0010h | |
| | CMD17 | 0000 0011h | |
| | CMD18 | 0000 0012h | |
| | CMD20 | 0000 0514h or 0000 0014h | |
| | CMD24 | 0000 0018h | |
| | CMD25 | 0000 0019h | |
| | CMD27 | 0000 001Bh | |
| | CMD28 | 0000 001Ch | |
| | CMD29 | 0000 001Dh | |
| | CMD30 | 0000 001Eh | |
| | CMD32 | 0000 0020h | |
| | CMD33 | 0000 0021h | |
| | CMD38 | 0000 0026h | |
| | CMD42 | 0000 002Ah | |
| | CMD52 | 0000 0434h or 0000 0034h | |
| | CMD53 | 0000 1C35h | Single block read |
| | | 0000 0C35h | Single block write |
| | | 0000 7C35h | Multi-block read |
| | | 0000 6C35h | Multi-block write |
| | | 0000 0035h | The setting to the left can be used regardless of the transfer being single block or multi-block. However, the MSB in the SDARG register (RW flag) must be set to 0 when reading and 1 when writing. |
| | CMD55 | 0000 0037h | |
| | CMD56 | 0000 0038h | |
| ACMD | ACMD6 | 0000 0046h | |
| | ACMD13 | 0000 004Dh | |
| | ACMD22 | 0000 0056h | |
| | ACMD23 | 0000 0057h | |
| | ACMD41 | 0000 0069h | |
| | ACMD42 | 0000 006Ah | |
| | ACMD51 | 0000 0073h | |

## 40.2.2 Argument Register (SDARG)

Address(es): SDHI.SDARG 0008 AC08h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The SDARG register is used for setting the argument field value. Set the SDARG register before setting the SDCMD register. The argument field value of the automatically issued CMD12 is 0000 0000h regardless of the SDARG register value.

## 40.2.3 Data Stop Register (SDSTOP)

Address(es): SDSTOP 0008 AC10h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SDBLKCNTEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | STP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | STP | Transfer Stop | Data transfer stops when this bit is set to 1. | R/W |
| b7 to b1 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b8 | SDBLKCNTEN | Block Count Register Value Select *1 | 0: SDBLKCNT register value is invalid<br>1: SDBLKCNT register value is valid | R/W |
| b31 to b9 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSTOP register stops data transfer. During a multi-block transfer sequence, the SDBLKCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SDSTOP register.

**STP Bit (Transfer Stop)**

When setting the STP bit to 1, set it after the SDSTS1.RSPEND flag becomes 1; when setting the STP bit to 0, set it after the SDSTS1.ACEND flag becomes 1. After a command sequence is complete, the SDHI does not issue CMD12 and the SDSTS1.ACEND flag does not become 1 even if the STP bit is set to 1. When the SDHI is in the busy state after receiving the R1b response, the SDHI does not issue CMD12 even if the STP bit is 1, and after the SDHI is released from

the busy state, the SDSTS1.ACEND flag becomes 1.

- Performing a multi-block transfer
  When the STP bit is set to 1, the SDHI issues CMD12, and the command sequence is stopped. The SD buffer can be accessed even after the STP bit is set to 1, but a buffer access error occurs and the SDSTS2.ILW flag or SDSTS2.ILR flag becomes 1. If the command sequence stops due to a communication error or a timeout, the SDHI does not issue CMD12.
- Performing a single block transfer
  When the STP bit is set to 1 during a single block write access, if there is no data in the SD buffer, the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer, after the SDHI is released from the busy state, the SDSTS1.ACEND flag becomes 1. When the STP bit is set to 1 during a single block read access, the SDSTS1.ACEND flag becomes 1. Also, CMD12 is not issued even if the STP bit is set to 1 during a single block read access or single block write access.

**SDBLKCNTEN Bit (Block Count Register Value Select)**

If the SDBLKCNTEN bit is 1 during a multi-block transfer sequence, the SDHI automatically issues CMD12. When the SDCMD.RSPTP[2:0] bits are set to 000b and CMD18 or CMD25 is issued, or if the SDCMD.RSPTP[2:0] bits are set to 011b, 100b, 101b, 110b, or 111b and the SDCMD.TRSTP bit is 1 (multiple blocks transferred), if the SDCMD.CMD12AT[1:0] bits are 00b (CMD12 is automatically issued during multi-block transfer), and the number of transfer blocks reaches the value set in the SDBLKCNT register, the SDHI automatically issues CMD12.
If the command sequence is stopped by a communication error or a timeout, CMD12 is not automatically issued.

## 40.2.4 Block Count Register (SDBLKCNT)

Address(es): SDHI.SDBLKCNT 0008 AC14h

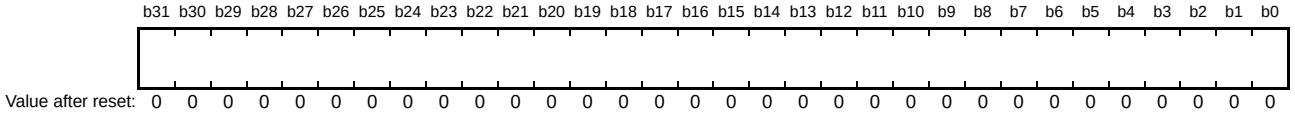
| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

When performing a multi-block transfer, SDBLKCNT is a readable/writable register used to set the number of blocks to be transferred. For example, when the register value is 0000 0001h, 1 block is transferred; when the register value is 0000 FFFFh, 65,535 blocks are transferred; and when the register value is FFFF FFFFh, 4,294,967,295 blocks are transferred. Do not set this register to 0000 0000h. Do not rewrite the SDBLKCNT register when the SDSTS2.CBSY flag is 1.

## 40.2.5 Response Register 10 (SDRSP10), Response Register 32 (SDRSP32), Response Register 54 (SDRSP54), Response Register 76 (SDRSP76)

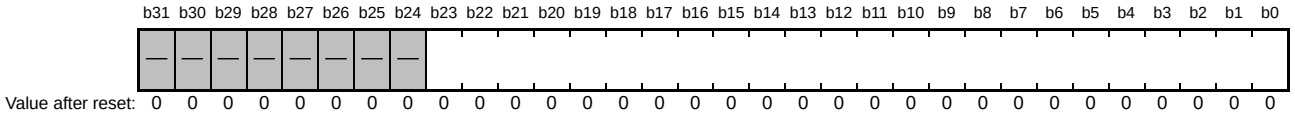
- SDRSP10, SDRSP32, SDRSP54

Address(es): SDHI.SDRSP10 0008 AC18h, SDHI.SDRSP32 0008 AC20h, SDHI.SDRSP54 0008 AC28h




- SDRSP76

Address(es): SDHI.SDRSP76 0008 AC30h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b23 to b0 | — | — | This register stores the response from the SD card. | R |
| b31 to b24 | — | Reserved | These bits are 0 when read. | R |

Registers SDRSP10, SDRSP32, SDRSP54, and SDRSP76 are read-only registers that store the response from the SD card. Depending on the type of response from the SD card, the SDHI divides and stores the response among the four registers.

Table 40.4 lists the correspondence between the response type and its storage destination.

**Table 40.4 Correspondence Between the Response Type and Its Storage Destination**

| Response Type | SDRSP10 Register | SDRSP32 Register | SDRSP54 Register | SDRSP76 Register |
|---|---|---|---|---|
| R1 | [39:8] | — | [39:8] *1 | — |
| R1b | [39:8] | — | [39:8] *1 | — |
| R2 | [39:8] | [71:40] | [103:72] | [127:104] |
| R3 | [39:8] | — | — | — |
| R4 | [39:8] | — | — | — |
| R5 | [39:8] | — | — | — |
| R6 | [39:8] | — | — | — |
| R7 | [39:8] | — | — | — |

Note 1. The response for CMD18 and CMD25 is stored in registers SDRSP10 and SDRSP54. Therefore, even if the SDRSP10 register is overwritten with the response for the automatically transmitted CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SDRSP54 register.

## 40.2.6 SD Status Register 1 (SDSTS1)

Address(es): SDHI.SDSTS1 0008 AC38h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SDD3MON | SDD3IN | SDD3RM |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | x | 0 | 0 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SDWPMON | — | SDCDMON | SDCDIN | SDCDRM | ACEND | — | RSPEND |
| Value after reset: | x | 0 | x | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RSPEND | Response End Detection Flag | 0: Response end is not detected<br>1: Response end is detected | R/(W) *1 |
| b1 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b2 | ACEND | Access End Detection Flag | 0: Access end is not detected<br>1: Access end is detected | R/(W) *1 |
| b3 | SDCDRM | SDHI_CD Removal Flag | 0: SD card removal not detected by the SDHI_CD pin<br>1: SD card removal detected by the SDHI_CD pin | R/(W) *1 |
| b4 | SDCDIN | SDHI_CD Insertion Flag | 0: SD card insertion not detected by the SDHI_CD pin<br>1: SD card insertion detected by the SDHI_CD pin | R/(W) *1 |
| b5 | SDCDMON | SDHI_CD Pin Monitor Flag | 0: SDHI_CD pin level is high *2<br>1: SDHI_CD pin level is low *2 | R |
| b6 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b7 | SDWPMON | SDHI_WP Pin Monitor Flag | 0: SDHI_WP pin level is high<br>1: SDHI_WP pin level is low | R |
| b8 | SDD3RM | SDHI_D3 Removal Flag | 0: SD card removal not detected by the SDHI_D3 pin<br>1: SD card removal detected by the SDHI_D3 pin | R/(W) *1 |
| b9 | SDD3IN | SDHI_D3 Insertion Flag | 0: SD card insertion not detected by the SDHI_D3 pin<br>1: SD card insertion detected by the SDHI_D3 pin | R/(W) *1 |
| b10 | SDD3MON | SDHI_D3 Pin Monitor Flag | 0: SDHI_D3 pin level is low<br>1: SDHI_D3 pin level is high | R |
| b31 to b11 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.
Note 2. The flag changes when the pin level continues for the period set in the SDOPT.CTOP[3:0] bits or longer.

The SDSTS1 register indicates the detection of a response end or access end for a command sequence. The SDSTS1 register also indicates the detection SD card insertion/removal, and indicates the write protection status.
During a multi-block transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag becomes 1, but the RSPEND flag remains set to 0.
If the command sequence is stopped due to a communication error or timeout, the ACEND flag or RSPEND flag becomes 1.
After a reset is released, the SDD3MON flag, SDD3IN flag, and SDD3RM flag values are changed according to the status of the SDHI_D3 pin, and their values are changed when data is being transferred in wide bus mode.
Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

### RSPEND Flag (Response End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A response is received.
- A command that does not have a response is issued.
- After the R1b response is received, the SDHI is released from the busy state.
- During a multi-block transmission, after the SDIOMD.C52PUB bit is set to 1, the CMD52 response is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: When a command is issued that is absent of data transfer, the RSPEND flag becomes 1 after the command sequence ends.

### ACEND Flag (Access End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block read sequence, the SD buffer read access is completed.
- During a multi-block read sequence, the last block is read from the SD buffer.
- During a multi-block read sequence, if CMD12 is automatically issued, data is read from the SD buffer, and the response for CMD12 is received.
- During a single block write sequence, after a CRC status token is received, the SDHI is released from the busy state.
- During a multi-block write sequence, after a CRC status token is received for the last block, the SDHI is released from the busy state.
- During a multi-block write sequence, when CMD12 is automatically issued, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response of the automatically issued CMD12 is received.
- During a multi-block write sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- During a multi-block write sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: The ACEND flag becomes 1 after the command sequence ends.

### SDCDRM Flag (SDHI_CD Removal Flag)

— This flag becomes 1 under the following condition:

- The SDHI_CD pin changes from low to high, and the high period is the period set in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

### SDCDIN Flag (SDHI_CD Insertion Flag)

— This flag becomes 1 under the following condition:

- The SDHI_CD pin changes from high to low, and the low period is the period specified in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

**SDD3RM Flag (SDHI_D3 Removal Flag)**

— This flag becomes 1 under the following condition:

- The SDHI_D3 pin changes from high to low, and the low period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

**SDD3IN Flag (SDHI_D3 Insertion Flag)**

— This flag becomes 1 under the following condition:

- The SDHI_D3 pin changes from low to high, and the high period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

## 40.2.7 SD Status Register 2 (SDSTS2)

Address(es): SDHI.SDSTS2 0008 AC3Ch

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | ILA | CBSY | SDCLKCREN | — | — | — | BWE | BRE |
| Value after reset: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SDD0MON | RSPTO | ILR | ILW | DTO | ENDE | CRCE | CMDE |
| Value after reset: | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMDE | Command Error Detection Flag | 0: Command error not detected<br>1: Command error detected | R/(W) *1 |
| b1 | CRCE | CRC Error Detection Flag | 0: CRC error not detected<br>1: CRC error detected | R/(W) *1 |
| b2 | ENDE | End Bit Error Detection Flag | 0: End bit error not detected<br>1: End bit error detected | R/(W) *1 |
| b3 | DTO | Data Timeout Detection Flag | 0: Data timeout not detected<br>1: Data timeout detected | R/(W) *1 |
| b4 | ILW | SDBUFR Illegal Write Access Detection Flag | 0: Illegal write access to the SDBUFR register not detected<br>1: Illegal write access to the SDBUFR register detected | R/(W) *1 |
| b5 | ILR | SDBUFR Illegal Read Access Detection Flag | 0: Illegal read access to the SDBUFR register not detected<br>1: Illegal read access to the SDBUFR register detected | R/(W) *1 |
| b6 | RSPTO | Response Timeout Detection Flag | 0: Response timeout not detected<br>1: Response timeout detected | R/(W) *1 |
| b7 | SDD0MON | SDHI_D0 Pin Status Flag | 0: SDHI_D0 pin is low<br>1: SDHI_D0 pin is high | R |
| b8 | BRE | SDBUFR Read Enable Flag | 0: Read access to the SDBUFR register disabled<br>1: Read access to the SDBUFR register enabled | R/(W) *1 |
| b9 | BWE | SDBUFR Write Enable Flag | 0: Write access to the SDBUFR register disabled<br>1: Write access to the SDBUFR register enabled | R/(W) *1 |
| b10 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b11 | — | Reserved | This bit is 0 when read. Set it to 1 when writing. | R/W |
| b12 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b13 | SDCLKCREN | SDCLKCR Write Enable Flag | 0: SD bus (CMD and DAT lines) is busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is disabled.<br>1: SD bus (CMD and DAT lines) is not busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is enabled. | R |
| b14 | CBSY | Command Sequence Status Flag | 0: Command sequence completed<br>1: Command sequence in progress (busy) | R |
| b15 | ILA | Illegal Access Error Detection Flag | 0: Illegal access error not detected<br>1: Illegal access error detected | R/(W) *1 |
| b31 to b16 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

The SDSTS2 register indicates the status of the SD buffer and the status of the SD card. Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

#### CMDE Flag (Command Error Detection Flag)

The command sequence is stopped when a command error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 40.3.6.8 or section 40.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The command index field value for the command transmitted differs from the command index field value for the response received.
- The command index field value for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) differs from the command index field value for the response received.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

#### CRCE Flag (CRC Error Detection Flag)

The command sequence is stopped when a CRC error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 40.3.6.8 or section 40.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The received CRC status token is in error (the value of the CRC status is a value other than 010b).
- The read data contains a CRC error.
- The response contains a CRC error.
- The response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains a CRC error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

#### ENDE Flag (End Bit Error Detection Flag)

The command sequence is stopped when an end bit error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 40.3.6.8 or section 40.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The response length is in error (the end bit could not be detected).
- The read data length is in error (the end bit of the enabled bit could not be detected).
- The CRC status token length is in error (the end bit could not be detected).
- The response length for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains an error (the end bit could not be detected).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

#### DTO Flag (Data Timeout Detection Flag)

This flag indicates that the data expected to be received during the period specified (set in the SDOPT.TOP[3:0] bits) was not received. However, response timeouts are excluded. The command sequence stops when a data timeout occurs.

— This flag becomes 1 under any of the following conditions:

- After the R1b response is received, the SDHI is busy for the period specified or longer.
- After the CRC status token is received, the SDHI is busy for the period specified or longer.
- After data is written, the CRC status token is not received even after the period specified elapsed.
- After a read command is issued, the read data is not received even after the period specified elapsed.
- After CMD12 is issued during a command sequence, the SDHI is busy for the period specified or longer.
- After the read data is received, the next read data is not received even after the period specified elapsed.
- After the SDHI exits the read wait state, the next read data is not received even after the period specified elapsed.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

#### ILW Flag (SDBUFR Illegal Write Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDBUFR register while the SDHI is not in the data read or data write command state.
- A value is written to the SDBUFR register while the SD buffer is full.
- A value is written to the SDBUFR register while the CRC status token or CRC status token length is in error.
- After the CRC status token is received, a value is written to the SDBUFR register if the SDHI is busy for the period set in bits SDOPT.TOP[3:0] or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

#### ILR Flag (SDBUFR Illegal Read Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- The SDBUFR register is read while the SD buffer is empty.
- The value read from the SDBUFR register includes a CRC error or end bit error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

#### RSPTO Flag (Response Timeout Detection Flag)

The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 40.3.6.8 or section 40.3.6.9 and complete the command sequence.

— This flag becomes 1 under the following condition:

- A response is not received even after 640 SDHI clock cycles or more have elapsed (including the response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer)).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

#### SDD0MON Flag (SDHI_D0 Pin Status Flag)

This flag indicates the status of the SDHI_D0 pin. After an erase command is issued, if the DTO flag is 1 and the RSPTO flag is 0, polling can be used to monitor the SDD0MON flag change from 0 to 1, and check that the erase command sequence is complete. If a communication error or timeout occurs during the write sequence, the SDHI_D0 pin may remain low. When the SDHI clock is stopped, the value before the SDHI clock was stopped is retained.

### BRE Flag (SDBUFR Read Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in the SD buffer.
- During a multi-block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in one of the two SD buffers.

— This flag becomes 0 under any of the following conditions:

- The bit is set to 0.
- DMA transfer is used to read 1 block of data from the SD buffer.

If the CPU is used to read data from the SDBUFR register, set the BRE flag to 0 before reading the data size *1 set in the SDSIZE.LEN[9:0] bits. Even if the block of data read contains a CRC error or end bit error, the data is stored in the SD buffer and the BRE flag becomes 1.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the odd numbered byte is ignored. Refer to section 40.5.2, SDBUFR Register Illegal Write Error for details.

### BWE Flag (SDBUFR Write Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the SD buffer is empty.
- During a multi-block transfer, bank 1 or bank 2 of the SD buffer is empty.

— This flag becomes 0 under any of the following conditions:

- The flag is set to 0.
- DMA transfer is used to write 1 block of data to the SD buffer.

If the CPU is used to write data to the SDBUFR register, set the BWE flag to 0 before writing the data size *1 set in the SDSIZE.LEN[9:0] bits.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the odd numbered byte is ignored. Refer to section 40.5.2, SDBUFR Register Illegal Write Error for details.

### SDCLKCREN Flag (SDCLKCR Write Enable Flag)

When a value is written to the SDCMD register, the SDHI starts the command sequence, the SDSTS2.CBSY flag becomes 1, and the SDSTS2.SDCLKCREN flag becomes 0. When the command sequence is complete, after the SDSTS2.CBSY flag becomes 0, eight cycles of the SDHI clock elapse and then the SDSTS2.SDCLKCREN flag becomes 1.

### ILA Flag (Illegal Access Error Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDCMD register when the SDSTS2.CBSY flag is 1.
- The SDCMD.CMDTP bit is set to 1 (command accompanying data transfer), the SDCMD.ACMD[1:0] bits are set to 00b, and the SDCMD.CMDIDX[5:0] bits are set to 001100b (CMD12).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

## 40.2.8 SD Interrupt Mask Register 1 (SDIMSK1)

Address(es): SDHI.SDIMSK1 0008 AC40h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | SDD3INM | SDD3RMM |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | SDCDINM | SDCDRMM | ACENDM | — | RSPENDM |
| Value after reset: | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RSPENDM | Response End Interrupt Request Mask | 0: Response end interrupt request is not masked<br>1: Response end interrupt request is masked | R/W |
| b1 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b2 | ACENDM | Access End Interrupt Request Mask | 0: Access end interrupt request is not masked<br>1: Access end interrupt request is masked | R/W |
| b3 | SDCDRMM | SDHI_CD Removal Interrupt Request Mask | 0: SD card removal interrupt request by the SDHI_CD pin not masked<br>1: SD card removal interrupt request by the SDHI_CD pin masked | R/W |
| b4 | SDCDINM | SDHI_CD Insertion Interrupt Request Mask | 0: SD card insertion interrupt request by the SDHI_CD pin not masked<br>1: SD card insertion interrupt request by the SDHI_CD pin masked | R/W |
| b7 to b5 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b8 | SDD3RMM | SDHI_D3 Removal Interrupt Request Mask | 0: SD card removal interrupt request by the SDHI_D3 pin not masked<br>1: SD card removal interrupt request by the SDHI_D3 pin masked | R/W |
| b9 | SDD3INM | SDHI_D3 Insertion Interrupt Request Mask | 0: SD card insertion interrupt request by the SDHI_D3 pin not masked<br>1: SD card insertion interrupt request by the SDHI_D3 pin masked | R/W |
| b31 to b10 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

The SDIMSK1 register enables and disables the interrupt requests from the status flags in the SDSTS1 register. Refer to Table 40.8, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

## 40.2.9 SD Interrupt Mask Register 2 (SDIMSK2)

Address(es): SDHI.SDIMSK2 0008 AC44h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | ILAM | — | — | — | — | — | BWEM | BREM |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | RSPTOM | ILRM | ILWM | DTTOM | ENDEM | CRCEM | CMDEM |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMDEM | Command Error Interrupt Request Mask | 0: Command error interrupt request not masked<br>1: Command error interrupt request masked | R/W |
| b1 | CRCEM | CRC Error Interrupt Request Mask | 0: CRC error interrupt request not masked<br>1: CRC error interrupt request masked | R/W |
| b2 | ENDEM | End Bit Error Interrupt Request Mask | 0: End bit detection error interrupt request not masked<br>1: End bit detection error interrupt request masked | R/W |
| b3 | DTTOM | Data Timeout Interrupt Request Mask | 0: Data timeout interrupt request not masked<br>1: Data timeout interrupt request masked | R/W |
| b4 | ILWM | SDBUFR Register Illegal Write Interrupt Request Mask | 0: Illegal write detection interrupt request for the SDBUFR register not masked<br>1: Illegal write detection interrupt request for the SDBUFR register masked | R/W |
| b5 | ILRM | SDBUFR Register Illegal Read Interrupt Request Mask | 0: Illegal read detection interrupt request for the SDBUFR register not masked<br>1: Illegal read detection interrupt request for the SDBUFR register masked | R/W |
| b6 | RSPTOM | Response Timeout Interrupt Request Mask | 0: Response timeout interrupt request not masked<br>1: Response timeout interrupt request masked | R/W |
| b7 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b8 | BREM | BRE Interrupt Request Mask | 0: Read enable interrupt request for the SDBUFR register not masked<br>1: Read enable interrupt request for the SDBUFR register masked | R/W |
| b9 | BWEM | BWE Interrupt Request Mask | 0: Write enable interrupt request for the SDBUFR register not masked<br>1: Write enable interrupt request for the SDBUFR register masked | R/W |
| b10 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b11 | — | Reserved | This bit is 1 when read and cannot be modified. | R |
| b14 to b12 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b15 | ILAM | Illegal Access Error Interrupt Request Mask | 0: Illegal access error interrupt request not masked<br>1: Illegal access error interrupt request masked | R/W |
| b31 to b16 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDIMSK2 register enables and disables the interrupt requests from the status flags in the SDSTS2 register. Refer to Table 40.8 for details on the relationship between the status flags and the requested interrupt source.

## 40.2.10 SDHI Clock Control Register (SDCLKCR)

Address(es): SDHI.SDCLKCR 0008 AC48h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CLKCTRLEN | CLKEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CLKSEL[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | CLKSEL[7:0] | SDHI Clock Frequency Select *1 | b7 b0<br>0 0 0 0 0 0 0 0: PCLKB divided by 2<br>0 0 0 0 0 0 0 1: PCLKB divided by 4<br>0 0 0 0 0 0 1 0: PCLKB divided by 8<br>0 0 0 0 0 1 0 0: PCLKB divided by 16<br>0 0 0 0 1 0 0 0: PCLKB divided by 32<br>0 0 0 1 0 0 0 0: PCLKB divided by 64<br>0 0 1 0 0 0 0 0: PCLKB divided by 128<br>0 1 0 0 0 0 0 0: PCLKB divided by 256<br>1 0 0 0 0 0 0 0: PCLKB divided by 512<br>1 1 1 1 1 1 1 1: PCLKB *2<br>Only set the values listed above. | R/W |
| b8 | CLKEN | SDHI Clock Output Control *1 | 0: SDHI clock output is disabled (SDHI_CLK signal fixed low)<br>1: SDHI clock output enabled | R/W |
| b9 | CLKCTRLEN | SDHI Clock Output Automatic Control Select | 0: Automatic control of SDHI clock output disabled<br>1: Automatic control of SDHI clock output enabled | R/W |
| b31 to b10 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SDSTS2.SDCLKCREN flag is 0.
Note 2. When setting the CLKSEL[7:0] bits to 11111111b or when changing the CLKSEL[7:0] bit values from 11111111b to another value, perform the following steps:
(1) Set the CLKEN bit to 0. Do not change the other bit values.
(2) Change the CLKSEL[7:0] bit values. Do not change the other bit values.
(3) Set the CLKEN bit to 1. Do not change the other bit values.

The SDCLKCR register controls the SDHI clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SDCMD register to start a command sequence. Do not write access the SDCLKCR register when the SDSTS2.SDCLKCREN flag is 0.

### CLKCTRLEN Bit (SDHI Clock Output Automatic Control Select)

The SDHI clock output automatic control is a function for starting and stopping SDHI clock output only during a command sequence. When this function is enabled, the SDHI starts outputting the SDHI clock after a value is set to the SDCMD register. After the command sequence is complete and eight cycles of the SDHI clock elapse, the SDHI stops outputting the SDHI clock. When the SDCLKCR.CLKEN bit is 0, output from the SDHI_CLK pin becomes low regardless of the CLKCTRLEN bit setting.

## 40.2.11 Transfer Data Size Register (SDSIZE)

Address(es): SDHI.SDSIZE 0008 AC4Ch

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | LEN[9:0] | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b9 to b0 | LEN[9:0] | Transfer Data Size Setting | Set the transfer data size. *1 | R/W |
| b11 to b10 | — | Reserved | These bits are 0 when read. Set them to 0 when writing. | R |
| b31 to b12 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SDSIZE register is used to set the transfer data size.

**LEN[9:0] Bits (Transfer Data Size Setting)**

When using single block transfer, the transfer data size can be set from 1 byte to 512 bytes. When CMD12 is automatically issued during a multi-block transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multi-block transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32-, 64-, 128-, or 256-byte multi-block read transfer can only be performed during an SDIO multi-block transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

## 40.2.12 Card Access Option Register (SDOPT)

Address SDHI.SDOPT 0008 AC50h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | WIDTH | — | — | — | — | — | — | — | TOP[3:0] | | | | CTOP[3:0] | | | |
| Value after reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | | R/W |
|---|---|---|---|---|---|
| b3 to b0 | CTOP[3:0] | Card Detection Time Counter *1 | b3 b0<br>0 0 0 0: PCLKB × $2^{10}$<br>0 0 0 1: PCLKB × $2^{11}$<br>0 0 1 0: PCLKB × $2^{12}$<br>0 0 1 1: PCLKB × $2^{13}$<br>0 1 0 0: PCLKB × $2^{14}$<br>0 1 0 1: PCLKB × $2^{15}$<br>0 1 1 0: PCLKB × $2^{16}$<br>0 1 1 1: PCLKB × $2^{17}$ | b3 b0<br>1 0 0 0: PCLKB × $2^{18}$<br>1 0 0 1: PCLKB × $2^{19}$<br>1 0 1 0: PCLKB × $2^{20}$<br>1 0 1 1: PCLKB × $2^{21}$<br>1 1 0 0: PCLKB × $2^{22}$<br>1 1 0 1: PCLKB × $2^{23}$<br>1 1 1 0: PCLKB × $2^{24}$<br>1 1 1 1: Do not set this value. | R/W |
| b7 to b4 | TOP[3:0] | Timeout Counter *1 | b7 b4<br>0 0 0 0: SDHI clock × $2^{13}$<br>0 0 0 1: SDHI clock × $2^{14}$<br>0 0 1 0: SDHI clock × $2^{15}$<br>0 0 1 1: SDHI clock × $2^{16}$<br>0 1 0 0: SDHI clock × $2^{17}$<br>0 1 0 1: SDHI clock × $2^{18}$<br>0 1 1 0: SDHI clock × $2^{19}$<br>0 1 1 1: SDHI clock × $2^{20}$ | b7 b4<br>1 0 0 0: SDHI clock × $2^{21}$<br>1 0 0 1: SDHI clock × $2^{22}$<br>1 0 1 0: SDHI clock × $2^{23}$<br>1 0 1 1: SDHI clock × $2^{24}$<br>1 1 0 0: SDHI clock × $2^{25}$<br>1 1 0 1: SDHI clock × $2^{26}$<br>1 1 1 0: SDHI clock × $2^{27}$<br>1 1 1 1: Do not set this value. | R/W |
| b8 | — | Reserved | This bit is 0 when read. Set it to 0 when writing. | | R/W |
| b12 to b9 | — | Reserved | These bits are 0 when read and cannot be modified. | | R |
| b13 | — | Reserved | This bit is 0 when read. Set it to 0 when writing. | | R/W |
| b14 | — | Reserved | This bit is 1 when read and cannot be modified. | | R |
| b15 | WIDTH | SD Bus Width Select *1 | 0: Wide bus mode (4 bits)<br>1: Default bus mode (1 bit) | | R/W |
| b31 to b16 | — | Reserved | These bits are 0 when read and cannot be modified. | | R |

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SD bus width and timeout counter are set in the SDOPT register.

## 40.2.13 SD Error Status Register 1 (SDERSTS1)

Address(es): SDHI.SDERSTS1 0008 AC58h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | — | CRCTK[2:0] | | | CRCTKE | RDCRCE | RSPCRCE1 | RSPCRCE0 |
| Value after reset: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | CRCLENE | RDLENE | RSPLENE1 | RSPLENE0 | CMDE1 | CMDE0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMDE0 | Command Error Flag 0 | 0: Command index field value for a command *1 response is error free<br>1: Command index field value for a command *1 response is in error | R |
| b1 | CMDE1 | Command Error Flag 1 | 0: Command index field value for a command *2 response is error free<br>1: Command index field value for a command *2 response is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the CMDE0 flag) | R |
| b2 | RSPLENE0 | Response Length Error Flag 0 | 0: Command *1 response length is error free<br>1: Command *1 response length is in error | R |
| b3 | RSPLENE1 | Response Length Error Flag 1 | 0: Command *2 response length is error free<br>1: Command *2 response length is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPLENE0 flag) | R |
| b4 | RDLENE | Read Data Length Error Flag | 0: Read data length error did not occur<br>1: Read data length error occurred | R |
| b5 | CRCLENE | CRC Status Token Length Error Flag | 0: CRC status token length error did not occur<br>1: CRC status token length error occurred | R |
| b7, b6 | — | Reserved | These bits are 0 when read. | R |
| b8 | RSPCRCE0 | Response CRC Error Flag 0 | 0: No CRC error detected in command *1 response<br>1: CRC error detected in command *1 response | R |
| b9 | RSPCRCE1 | Response CRC Error Flag 1 | 0: No CRC error detected in command *2 response<br>1: CRC error detected in command *2 response (the error that occurs for CMD12 with the setting of the SDCMD.CMDIDX[5:0] bits is indicated by the RSPCRCE0 flag) | R |
| b10 | RDCRCE | Read Data CRC Error Flag | 0: No CRC error detected in read data<br>1: CRC error detected in read data | R |
| b11 | CRCTKE | CRC Status Token Error Flag | 0: No error detected in CRC status token<br>1: Error detected in CRC status token | R |
| b14 to b12 | CRCTK[2:0] | CRC Status Token | Store the CRC status token value (normal value is 010b) | R |
| b15 | — | Reserved | This bit is 0 when read. | R |
| b31 to b16 | — | Reserved | These bits are undefined when read. | R |

Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop data transfer.

The SDERSTS1 register indicates the CRC status token, CRC error, end bit error, and command error.

## 40.2.14 SD Error Status Register 2 (SDERSTS2)

Address(es): SDHI.SDERSTS2 0008 AC5Ch

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | CRCBSYTO | CRCTO | RDTO | BSYTO1 | BSYTO0 | RSPTO1 | RSPTO0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RSPTO0 | Response Timeout Flag 0 | 0: After a command *1 was issued, a response was received in less than 640 cycles of the SDHI clock.<br>1: After a command *1 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed. | R |
| b1 | RSPTO1 | Response Timeout Flag 1 | 0: After a command *2 was issued, a response was received in less than 640 cycles of the SDHI clock.<br>1: After a command *2 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPTO0 flag). | R |
| b2 | BSYTO0 | Busy Timeout Flag 0 | 0: After the R1b response was received, the SDHI was released from the busy state during the specified period *3.<br>1: After the R1b response was received, the SDHI was in the busy state even after the specified period *3 elapsed. | R |
| b3 | BSYTO1 | Busy Timeout Flag 1 | 0: After CMD12 was automatically issued, the SDHI was released from the busy state during the specified period *3.<br>1: After CMD12 was automatically issued, the SDHI was in the busy state even after the specified period *3 elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the BSYTO0 flag). | R |
| b4 | RDTO | Read Data Timeout Flag | After a read command is issued, this flag becomes 1 when read data is not received even after the specified period *3 elapses.<br>After read data is received, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses.<br>After the SDHI exits the read wait state, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses. | R |
| b5 | CRCTO | CRC Status Token Timeout Flag | 0: After data was written to the SD card, a CRC status token was received during the specified period *3.<br>1: After CRC data was written to the SD card, a CRC status token was not received even after the specified period *3 elapsed. | R |
| b6 | CRCBSYTO | CRC Status Token Busy Timeout Flag | 0: After a CRC status token was received, the SDHI was released from the busy state during the specified period *3.<br>1: After a CRC status token was received, the SDHI is in the busy state even after the specified period *3 elapsed. | R |
| b31 to b7 | — | Reserved | These bits are 0 when read. | R |

Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop transfer.

Note 3. Set the SDOPT.TOP[3:0] bits to select the number of *n* cycles.

The SDERSTS2 register indicates the timeout status.

## 40.2.15 SD Buffer Register (SDBUFR)

Address(es): SDHI.SDBUFR 0008 AC60h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | Undefined | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The SDBUFR register is used when writing data to the SD card and when reading data from the SD card. The SDBUFR register is connected to the SDHI's internal SD buffer. Refer to section 40.3.1, Data Block Format of the SD Card for details on the configuration of the SDBUFR register and the SD buffer.

## 40.2.16 SDIO Mode Control Register (SDIOMD)

Address(es): SDHI.SDIOMD 0008 AC68h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | C52PUB | IOABT | — | — | — | — | — | RWREQ | — | INTEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | INTEN | SDIO Interrupt Acceptance Enable *1 | 0: SDIO interrupt accept disabled<br>1: SDIO interrupt accept enabled | R/W |
| b1 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b2 | RWREQ | Read Wait Request | 0: SDHI exits read wait state<br>1: Request for SDHI to enter read wait state | R/W |
| b7 to b3 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b8 | IOABT | SDIO Abort | If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted. | R/W |
| b9 | C52PUB | SDIO None Abort | If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is issued before the transfer process is complete, and the command sequence is completed. | R/W |
| b31 to b10 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDIOMD register controls reception of the SDIO interrupt, controls CMD52 issuance during multi-block transfer, and controls the read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

### RWREQ Bit (Read Wait Request)

If the RWREQ bit is set to 1 during a multi-block read sequence triggered by issuing CMD53, when the current block is done being read, the SDHI enters the read wait state. The method for exiting the read wait state is as follows.

- If the RWREQ bit is set to 0 while the SDHI is in the read wait state, the SDHI exits the read wait state.
- If the IOABT bit is set to 1 while the SDHI is in the read wait state, after CMD52 is issued, the RWREQ bit becomes 0 and the SDHI exits the read wait state.
- If bits C52PUB and RWREQ are simultaneously set to 1 during a multi-block read sequence triggered by issuing CMD53 *1, the SDHI does not automatically exit the read wait state, so after receiving the CMD52 response, set the RWREQ bit to 0.

Note 1. Set bits RWREQ and C52PUB to 1 simultaneously.

If the RWREQ bit is set to 1 while the last block is being transferred during a multi-block read sequence triggered by issuing CMD53, the SDHI will not enter the read wait state, the SDSTS1.ACEND flag becomes 1, and the RWREQ bit becomes 0. Set the RWREQ bit to 1 after the SDSTS1.RSPEND flag becomes 1.

### IOABT Bit (SDIO Abort)

- If the IOABT bit is set to 1 during a multi-block transfer sequence triggered by issuing CMD53, the SDHI stops the CMD53 command sequence, and CMD52 is issued. If the command sequence is stopped due to a communication error or timeout, the SDHI does not issue CMD52. The SD buffer can be accessed even after the IOABT bit is set to 1, but the SDSTS2.ILR flag or ILW flag becomes 1, and a buffer access error occurs. Write a value to the SDARG register before setting the IOABT bit to 1.
- During a single block write, if there is no data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 during a single block read, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag immediately becomes 1.
- If the SDHI is in the busy state after the R1b response is received and the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 after the command sequence is completed, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag does not become 1.
- Set the IOABT bit to 1 after the SDSTS1.RSPEND flag becomes 1.
- Set the IOABT bit to 0 after the SDSTS1.ACEND flag becomes 1.

### C52PUB Bit (SDIO None Abort)

- If the C52PUB bit is set to 1 during a multi-block write sequence triggered by issuing CMD53, CMD52 is automatically issued when the SD buffer is empty and the current block write access is complete. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- If the C52PUB bit and RWREQ bit are set to 1 during a multi-block read sequence triggered by issuing CMD53, the SDHI enters the read wait state after the current block read access is complete, and the SDHI automatically issues CMD52. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is set to 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- During a multi-block read sequence triggered by issuing CMD53, if the C52PUB bit is set to 1, also set the RWREQ bit to 1.
- Write a value to the SDARG register before setting the C52PUB bit to 1.
- Set the C52PUB bit to 1 after the SDSTS1.RSPEND flag becomes 1.

## 40.2.17 SDIO Status Register (SDIOSTS)

Address(es): SDHI.SDIOSTS 0008 AC6Ch

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | EXWT | EXPUB52 | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | IOIRQ |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IOIRQ | SDIO Interrupt Status Flag | 0: SDIO interrupt not accepted<br>1: SDIO interrupt accepted | R/(W) *1 |
| b2, b1 | — | Reserved | These bits are undefined when read. Set them to 1 when writing. | R/W |
| b13 to b3 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b14 | EXPUB52 | EXPUB52 Status Flag | Indicates the status of the EXPUB52 | R/(W) *1 |
| b15 | EXWT | EXWT Status Flag | Indicates the status of the EXWT | R/(W) *1 |
| b31 to b16 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. The flag value does not change even when set to 1. If 0 is written to this flag, it becomes 0.

The SDIOSTS register indicates the status of the SDIO card access. When clearing a flag, bits to be cleared should be set to 0; bits not being cleared should be set to 1.

### IOIRQ Flag (SDIO Interrupt Status Flag)

— This flag becomes 1 under the following condition:

- The SDIO interrupt from the SDIO card is accepted while the SDIOMD.INTEN bit is 1.

— This flag becomes 0 under the following condition:

- The flag is set to 0. *1

Note 1. Access the SDIO card, negate the SDIO interrupt from the SDIO card, and then set the IOIRQ flag to 0. If the SDIO interrupt from the SDIO card is not negated, the IOIRQ flag might become 1 again.

### EXPUB52 Flag (EXPUB52 Status Flag)

— This flag becomes 1 under any of the following conditions:

- When multi-block transfer is triggered by CMD53 being issued, the SDIOMD.C52PUB bit is set to 1 while the last block is being transferred.
- When multi-block write is triggered by CMD53 being issued, the SDIOMD.C52PUB bit remains set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

**EXWT Flag (EXWT Status Flag)**

— This flag becomes 1 under the following condition:

- During a multi-block read sequence triggered by CMD53 being issued, the SDIOMD.RWREQ bit is set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

## 40.2.18 SDIO Interrupt Mask Register (SDIOIMSK)

Address(es): SDHI.SDIOIMSK 0008 AC70h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
|---|---|---|---|---|---|---|---|---|
| | EXWTM | EXPUB52M | — | — | — | — | — | — |
| Value after reset: | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | IOIRQM |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IOIRQM | IOIRQ Interrupt Mask Control | 0: IOIRQ interrupt not masked<br>1: IOIRQ interrupt masked | R/W |
| b2, b1 | — | Reserved | These bits are 1 when read. Set them to 1 when writing. | R/W |
| b13 to b3 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b14 | EXPUB52M | EXPUB52 Interrupt Request Mask Control | 0: EXPUB52 interrupt request not masked<br>1: EXPUB52 interrupt request masked | R/W |
| b15 | EXWTM | EXWT Interrupt Request Mask Control | 0: EXWT interrupt request not masked<br>1: EXWT interrupt request masked | R/W |
| b31 to b16 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

The SDIOIMSK register enables and disables the interrupt requests from the status flags in the SDIOSTS register. Refer to Table 40.8, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

## 40.2.19 DMA Transfer Enable Register (SDDMAEN)

Address(es): SDHI.SDDMAEN 0008 ADB0h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DMAEN | — |
| Value after reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | — | Reserved | This bit is 0 when read. Set it to 0 when writing. | R/W |
| b1 | DMAEN | DMA Transfer Enable *1 *2 | 0: Using DMAC and DTC to access the SDBUFR register is disabled<br>1: Using DMAC and DTC to access the SDBUFR register is enabled | R/W |
| b3, b2 | — | Reserved | These bits are 0 when read. Set them to 0 when writing. | R |
| b4 | — | Reserved | This bit is 1 when read. Set it to 1 when writing. | R |
| b5 | — | Reserved | This bit is 0 when read. Set it to 0 when writing. | R/W |
| b7, b6 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b9, b8 | — | Reserved | These bits are 0 when read. Set them to 0 when writing. | R/W |
| b11, b10 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b12 | — | Reserved | This bit is 1 when read. Set it to 1 when writing. | R |
| b31 to b13 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. Do not rewrite this bit when the SDSTS2.CBSY bit is 1.
Note 2. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDDMAEN register enables and disables DMA transfer.

### DMAEN Bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SDCMD register.

## 40.2.20 SDHI Software Reset Register (SDRST)

Address(es): SDHI.SDRST 0008 ADC0h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SDRST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SDRST | SDHI Software Reset Control | 0: SDHI software reset<br>1: SDHI software reset released | R/W |
| b2, b1 | — | Reserved | These bits are 1 when read. Set them to 1 when writing. | R |
| b31 to b3 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Table 40.5 lists the bits and flags initialized by the SDHI software reset.

**Table 40.5 Bits and Flags Initialized by the SDHI Software Reset**

| Register | Bit/Flag |
|---|---|
| SDSTOP | SDBLKCNTEN |
| SDSTS1 | RSPEND, ACEND |
| SDSTS2 | CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, SDD0MON, BRE, BWE, SDCLKCREN, ILA |
| SDCLKCR | CLKEN |
| SDOPT | CTOP[3:0], TOP[3:0], WIDTH<br>Bits b8 and b13 in the SDOPT register are also initialized by the SDHI software reset. |
| SDERSTS1 | CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0] |
| SDERSTS2 | RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO |
| SDIOSTS | IOIRQ, EXPUB52, EXWT |

## 40.2.21 Swap Control Register (SDSWAP)

Address(es): SDHI.SDSWAP 0008 ADE0h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | BRSWP | BWSWP | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b1 | — | Reserved | This bit is 0 when read. Set it to 0 when writing. | R/W |
| b2 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b4, b3 | — | Reserved | These bits are 0 when read. Set them to 0 when writing. | R/W |
| b5 | — | Reserved | This bit is 0 when read and cannot be modified. | R |
| b6 | BWSWP | SDBUFR Swap Write *1 | 0: Normal write operation<br>1: Swap the byte endian before writing to the SDBUFR register | R/W |
| b7 | BRSWP | SDBUFR Swap Read *1 | 0: Normal read operation<br>1: Swap the byte endian before reading the SDBUFR register | R/W |
| b10 to b8 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b12, b11 | — | Reserved | These bits are 0 when read. Set them to 0 when writing. | R/W |
| b14, b13 | — | Reserved | These bits are 0 when read and cannot be modified. | R |
| b15 | — | Reserved | This bit is 0 when read. Set it to 0 when writing. | R/W |
| b31 to b16 | — | Reserved | These bits are 0 when read and cannot be modified. | R |

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSWAP register is used to select whether or not the byte endian is swapped when accessing the SDBUFR register. Refer to section 40.3.1 for details on the differences in accessing the SDBUFR register based on the SDSWAP register value.

## 40.3 SDHI Operation

### 40.3.1 Data Block Format of the SD Card

The SDHI has a default bus mode (1-bit width) that uses just the SDHI_D0 pin as a data line and a wide bus mode (4-bit width) that uses pins SDHI_D0 to SDHI_D3. Figure 40.2 shows the transfer format when the SDOPT.WIDTH bit is 1 (default bus mode), and Figure 40.3 shows the transfer format when the SDOPT.WIDTH bit is 0 (wide bus mode).

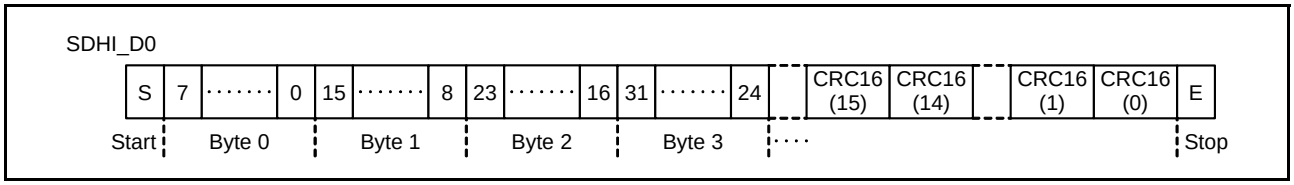



**Figure 40.2 Transfer Format in Default Bus Mode**

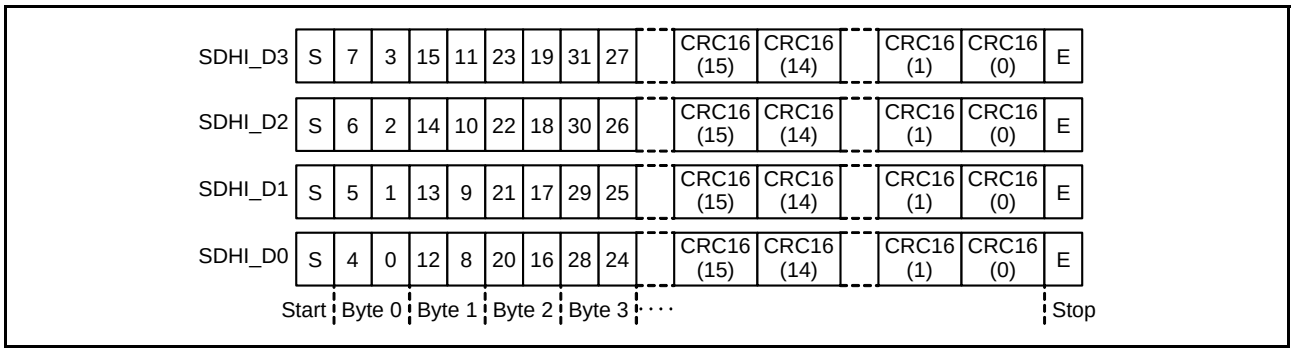



**Figure 40.3 Transfer Format in Wide Bus Mode**

### 40.3.2 SD Buffer and the SDBUFR Register

The SDHI transfers data to an SD card via its internal SD buffer. The SD buffer is comprised of a double buffer, and each buffer is 512 bytes. Figure 40.4 shows the data configuration of a single buffer of the SD buffer's double buffer.

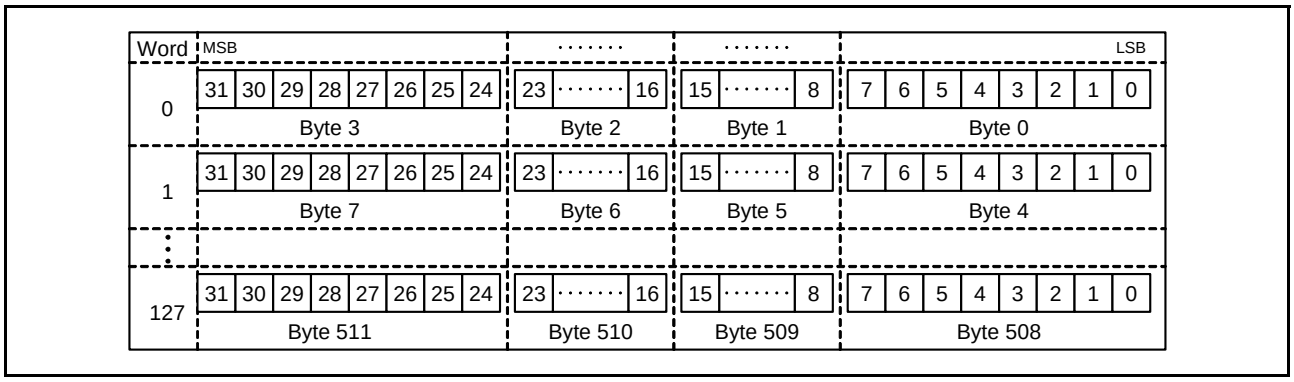



**Figure 40.4 Data Configuration of Single Buffer in the SD Buffer**

Access to the SD buffer is done via the SDBUFR register. If data is written to the SDBUFR register while the SDSWAP.BWSWP bit is 1, the SDHI swaps the endian for the byte, and stores the data in the SDBUFR register. If data is read from the SDBUFR register while the SDSWAP.BRSWP bit is 1, the data of the endian for the byte that was swapped can be read. Figure 40.5 shows the data alignment when reading the SDBUFR register.

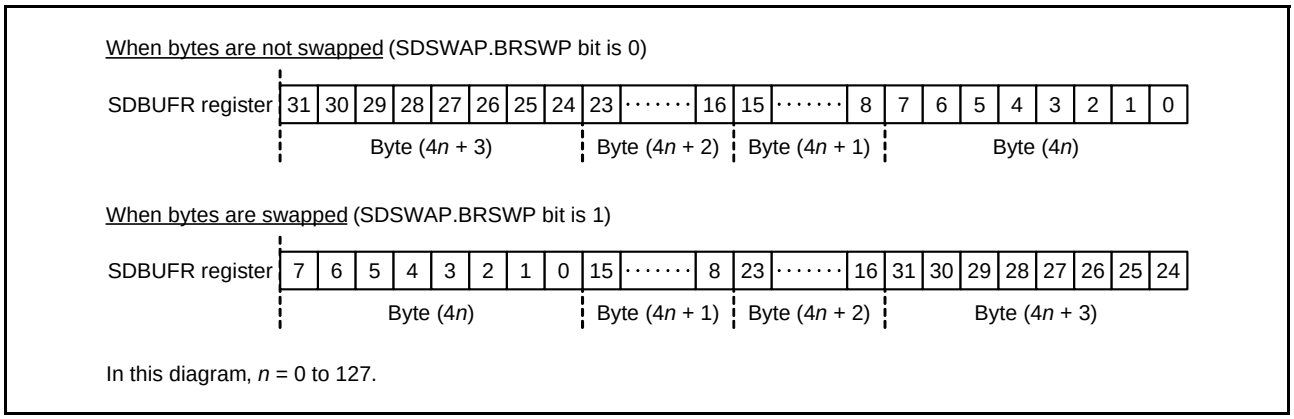



**Figure 40.5 Data Alignment When Reading the SDBUFR Register**

### 40.3.3 SD Card Detection

The SDHI can detect an SD card using either the SDHI_CD pin or SDHI_D3 pin.

#### 40.3.3.1 Using the SDHI_CD Pin to Detect an SD Card

Figure 40.6 shows the timing chart for SD card detection using the SDHI_CD pin. The SDHI_CD pin is connected to the card detection switch of the SD card connector, and is pulled-up by the MCU. The pull-up resistance value is determined by the specifications of the host device. Note that there are some SD card sockets whose card detection switches become open when the SD card is inserted.

- Detecting SD card insertion
  The signal from the SDHI_CD pin becomes low when an SD card is inserted. This causes the SDSTS1.SDCDIN flag to become 1 if the SDHI_CD pin is low for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDIN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SDHI_CD pin becomes high when the SD card is removed. This causes the SDSTS1.SDCDRM flag to become 1 if the SDHI_CD pin is high for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDRM flag is cleared by setting it to 0.

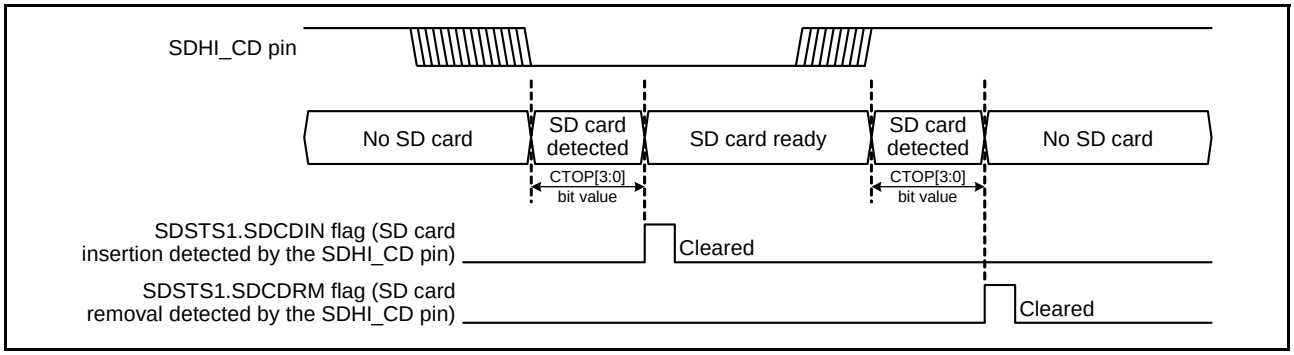



**Figure 40.6 SD Card Detection Using the SDHI_CD Pin**

#### 40.3.3.2 Using the SDHI_D3 Pin to Detect an SD Card

Figure 40.7 shows the timing chart for SD card detection using the SDHI_D3 pin. The SDHI_D3 pin is pulled-down by the MCU. The pull-down resistance value is determined by the specifications of the host device.

- Detecting SD card insertion
  The signal from the SDHI_D3 pin becomes high when an SD card is inserted. This causes the SDSTS1.SDD3IN flag to become 1. The SDSTS1.SDD3IN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SDHI_D3 pin becomes low when the SD card is removed. This causes the SDSTS1.SDD3RM flag to become 1. The SDSTS1.SDD3RM flag is cleared by setting it to 0.

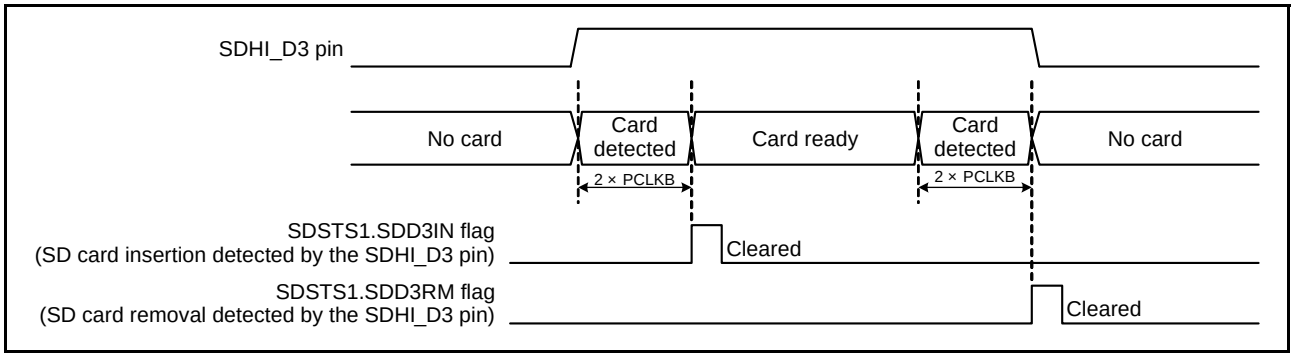



**Figure 40.7 SD Card Detection Using the SDHI_D3 Pin**

### 40.3.4 SD Card Write Protection

The SDHI can disable writing to an SD card via the SDHI_WP pin or a command.

#### 40.3.4.1 Using the SDHI_WP Pin to Enable Write Protection

The SDHI_WP pin is connected to the WP detection switch of the SD card connector, and the SDHI_WP pin is pulled-down or pulled-up when an SD card is inserted. The resistance value and whether the SDHI_WP pin is pulled-up or pulled-down are determined by the specifications of the host device. The status of the SDHI_WP pin is reflected in the SDSTS1.SDWPMON flag. After an SD card is inserted, read the SDSTS1.SDWPMON flag to check if write protection is enabled or disabled.

#### 40.3.4.2 Using a Command to Enable Write Protection

The SDHI uses the write protect command or the SD card lock command to disable writing to the SD card.

### 40.3.5 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the corresponding status flag in the SDSTS2 register becomes 1. Also, depending on the source of the error, the corresponding flag in the SDERSTS1 or SDERSTS2 register becomes 1.

The status flags in registers SDERSTS1 and SDERSTS2 become 0 by writing to the SDCMD register, or by setting the SDRST.SDRST bit to 0.

**Table 40.6 Communication Errors**

| Communication Error | Interrupt Flag Register | | Error Status Register | | This Occurs When... |
|---|---|---|---|---|---|
| | Register symbol | Bit symbol | Register symbol | Bit symbol | |
| End bit error | SDSTS2 | ENDE | SDERSTS1 | CRCLENE | The CRC status token length is in error |
| | | | | RDLENE | The read data length is in error |
| | | | | RSPLENE1 | The response length is in error *1 |
| | | | | RSPLENE0 | The response length is in error *2 |
| CRC error | | CRCE | | CRCTKE | The CRC status token is in error |
| | | | | RDCRCE | There is a CRC error in the read data |
| | | | | RSPCRCE1 | There is a CRC error in the response *1 |
| | | | | RSPCRCE0 | There is a CRC error in the response *2 |
| Command error | | CMDE | | CMDE1 | The command index field value for the transmitted command and received response do not match *1 |
| | | | | CMDE0 | The command index field value for the transmitted command and received response do not match *2 |

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.

**Table 40.7 Timeouts**

| Timeout | Interrupt Flag Register | | Error Status Register | | This Occurs When... |
|---|---|---|---|---|---|
| | Register symbol | Bit symbol | Register symbol | Bit symbol | |
| Response timeout | SDSTS2 | RSPTO | SDERSTS2 | RSPTO1 | A response is not received even after a minimum of 640 SDHI clock cycles elapse *1 |
| | | | | RSPTO0 | A response is not received even after a minimum of 640 SDHI clock cycles elapse *2 |
| Data timeout (excluding response timeout) | | DTO | | CRCBSYTO | After the CRC status token is received, the SDHI is busy for at least the period set *3 |
| | | | | CRCTO | After the write data is transmitted, the CRC status token is not received even after at least the period set *3 elapses |
| | | | | RDTO | After the read command is issued, the read data is not received even after at least the period set *3 elapses |
| | | | | | After the read data is received, the next block read data is not received even after at least the period set *3 elapses |
| | | | | | After the SDHI exits the read wait state, the next block read data is not received even after at least the period set *3 elapses |
| | | | | BSYTO1 | After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set *3 |
| | | | | BSYTO0 | After the R1b response is received, the SDHI is busy for at least the period set *3 (a command other than CMD12 is issued during the command sequence) |

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.
Note 3. The period is set in the SDOPT.TOP[3:0] bits.

## 40.3.6 Examples of Issuing a Command

### 40.3.6.1 Command Absent of Response Reception and Data Transfer

Figure 40.8 shows an example of no response being received and no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value to the SDARG register, write the command information to be sent to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a command is issued, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0.

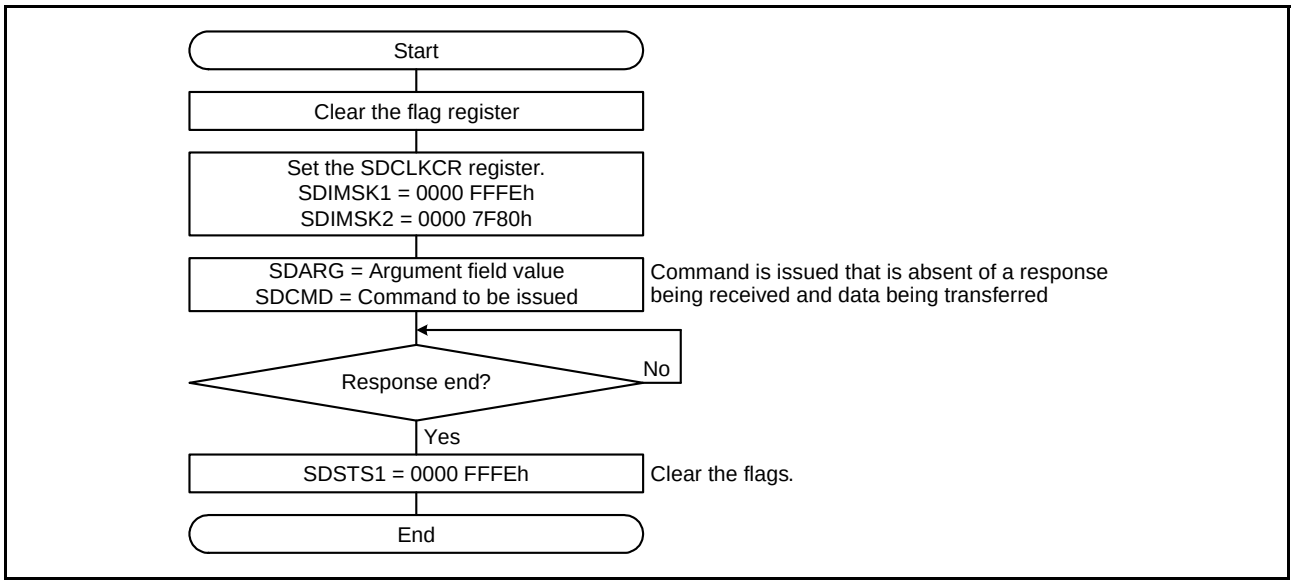



**Figure 40.8 Command Issued That Is Absent of Response Reception and Data Transfer**

### 40.3.6.2 Command Absent of Data Transfer

Figure 40.9 shows an example of no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value in the SDARG register, write the information of the command to be issued to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

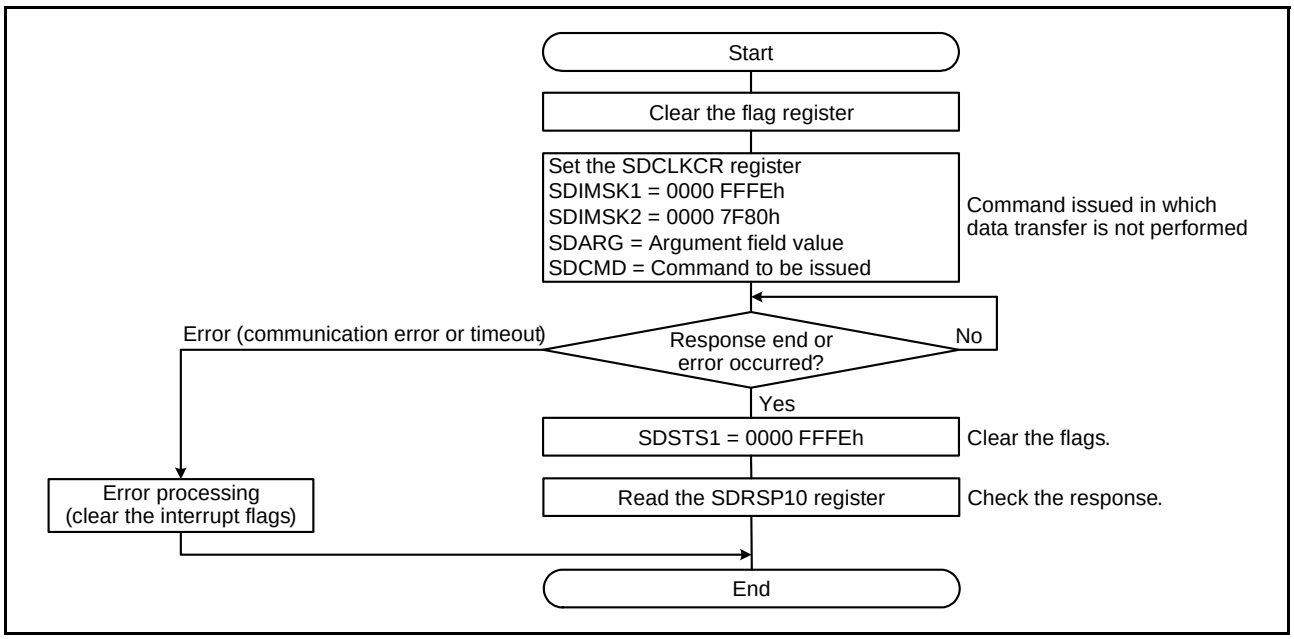



**Figure 40.9 Command Issued That Is Absent of Data Transfer**

#### 40.3.6.3 Single Block Read Command (CMD17)

Figure 40.10 shows an example of issuing the single block read command (CMD17).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD17 to the SDARG register, write 0000 0011h to the SDCMD register. The SDHI issues CMD17 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register. If the response read is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that CMD12 and CMD52 are not automatically issued by stopping this command sequence.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0 and set the SDIMSK2.BREM bit to 0.
7. After the amount of data set in the SDSIZE.LEN[9:0] bits is received from the SD card, the SDSTS2.BRE flag becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register.
9. After data has been read from the SDBUFR register, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

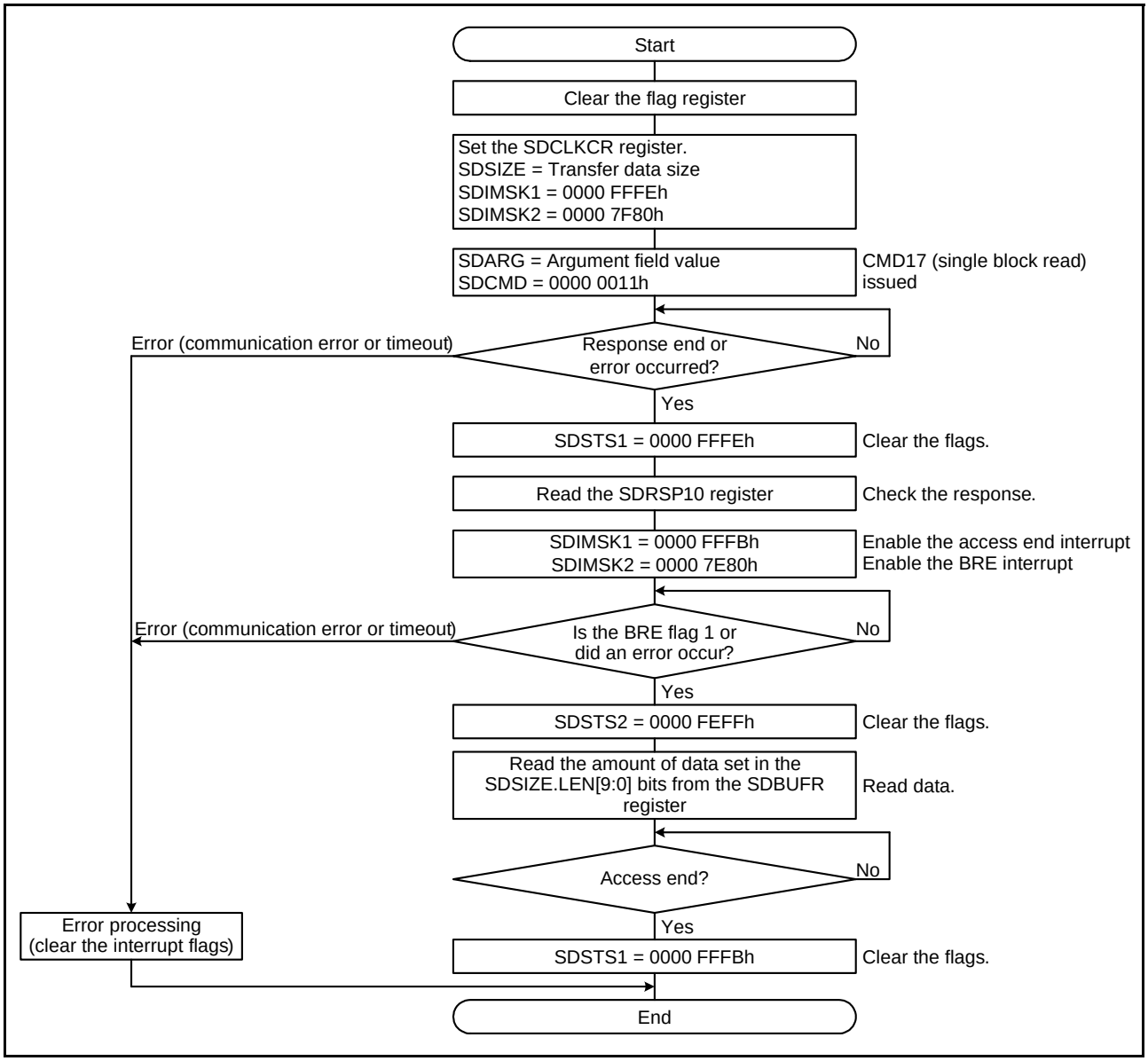



**Figure 40.10 Issuing the Single Block Read Command**

#### 40.3.6.4 Single Block Write Command (CMD24)

Figure 40.11 shows an example of issuing the single block write command (CMD24).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD24 to the SDARG register, write 0000 0018h to the SDCMD register. The SDHI issues CMD24 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP10 register. If the read response is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that when this command sequence is stopped, CMD12 and CMD52 are not automatically issued.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BWEM bit to 0.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, the SDHI transmits write data to the SD card. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur.
9. After all data has been written to the SD card, the SDHI receives the CRC status token, and the SDHI_D0 pin line becomes busy (low). Then, when the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

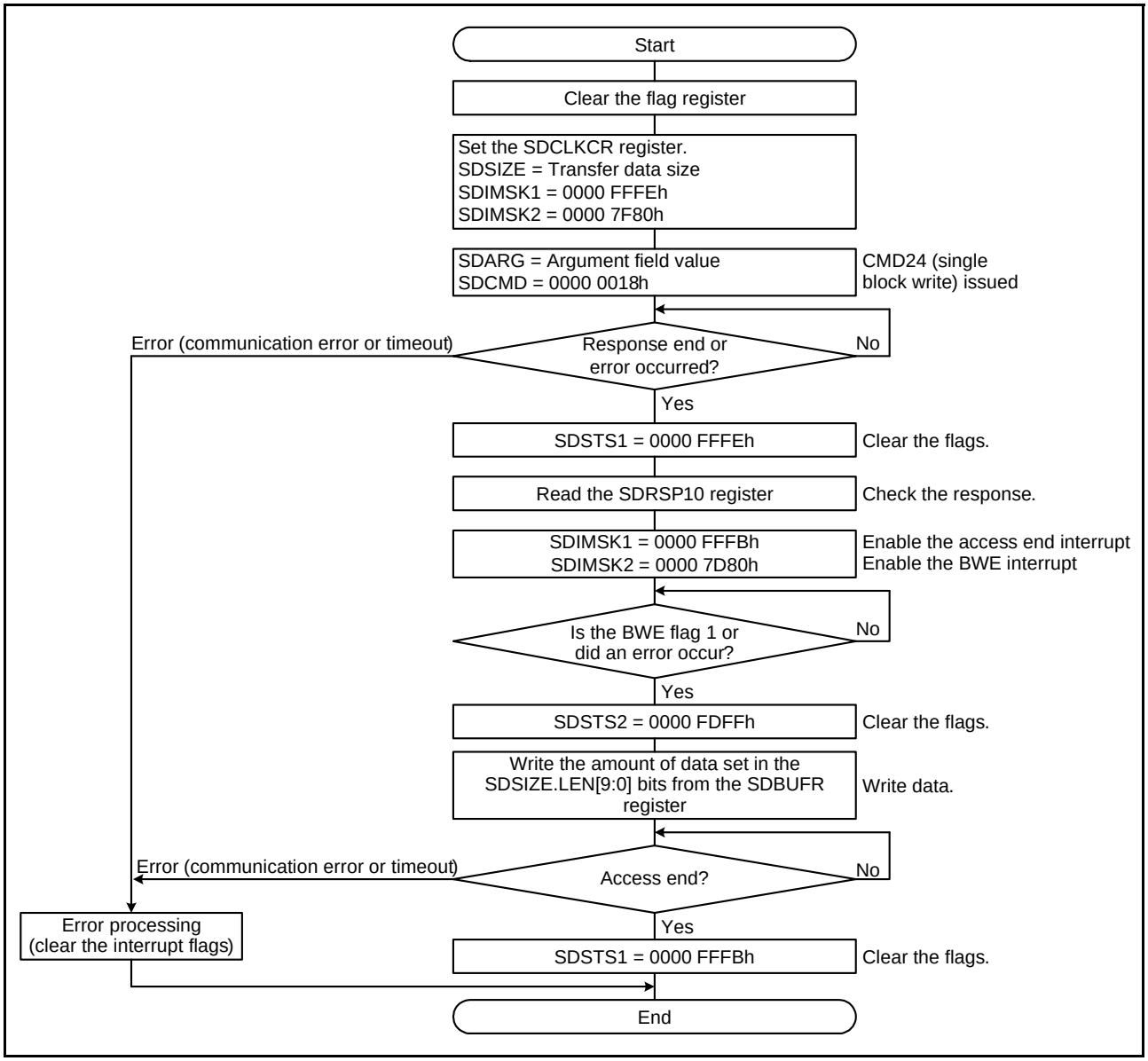



**Figure 40.11 Issuing the Single Block Write Command**

#### 40.3.6.5 Multi-Block Read Command (CMD18)

Figure 40.12 shows an example of issuing the multi-block read command (CMD18).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD18 to the SDARG register, write 0000 0012h to the SDCMD register. The SDHI issues CMD18 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BREM bit to 0.
7. After receiving one block of data from the SD card, the SDSTS2.BRE bit becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register. The read access to the SDBUFR register repeats for the amount of transfer blocks set in the SDBLKCNT register. Also, while reading the SDBUFR register, data reception may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been read, the SDHI automatically issues CMD12, and the response is received. At this time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been received and the CMD12 response is received, the SDSTS1.ACEND flag becomes 1 and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

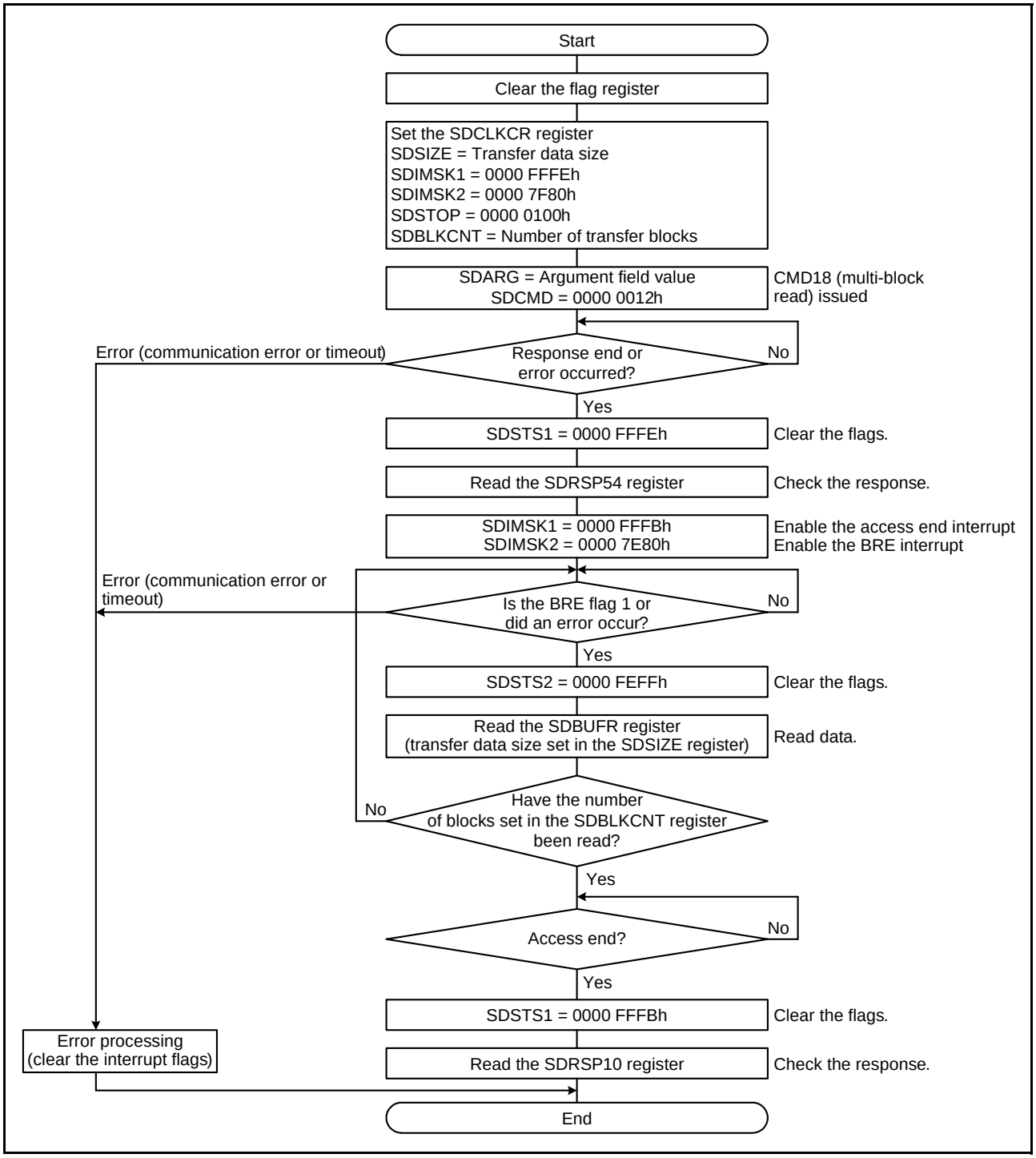



**Figure 40.12 Issuing the Multi-Block Read Command**

### 40.3.6.6 Multi-Block Write Command (CMD25)

Figure 40.13 shows an example of issuing the multi-block write command (CMD25).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 40.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD25 to the SDARG register, write 0000 0019h to the SDCMD register. The SDHI issues CMD25 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, configure the SDIMSK1 register to enable the access end interrupt request, and configure the SDIMSK2 register to enable the BWE interrupt request.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, and after the SDHI transmits write data to the SD card, the CRC status token is received, and the SDHI_D0 pin line becomes busy (low). The write access to the SDBUFR register and CRC status token reception repeat for the amount of transfer blocks set in the SDBLKCNT register. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been written, the SDHI automatically issues CMD12, and the response is received. At that time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been transmitted and the CRC status token is received, the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

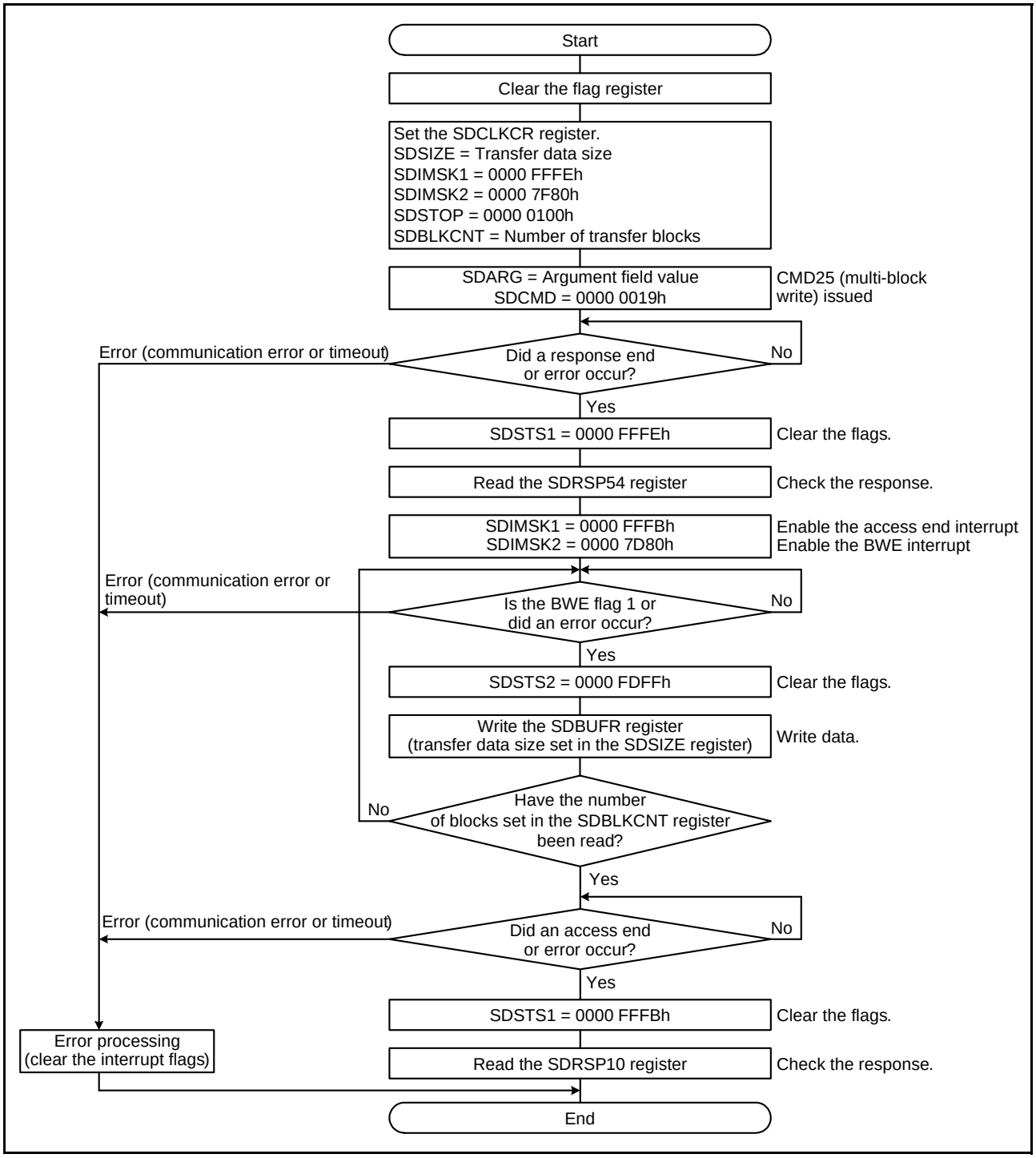



**Figure 40.13 Issuing the Multi-Block Write Command**

### 40.3.6.7 IO_RW_DIRECT Command (CMD52)

Figure 40.14 shows an example of issuing the IO_RW_DIRECT command (CMD52).

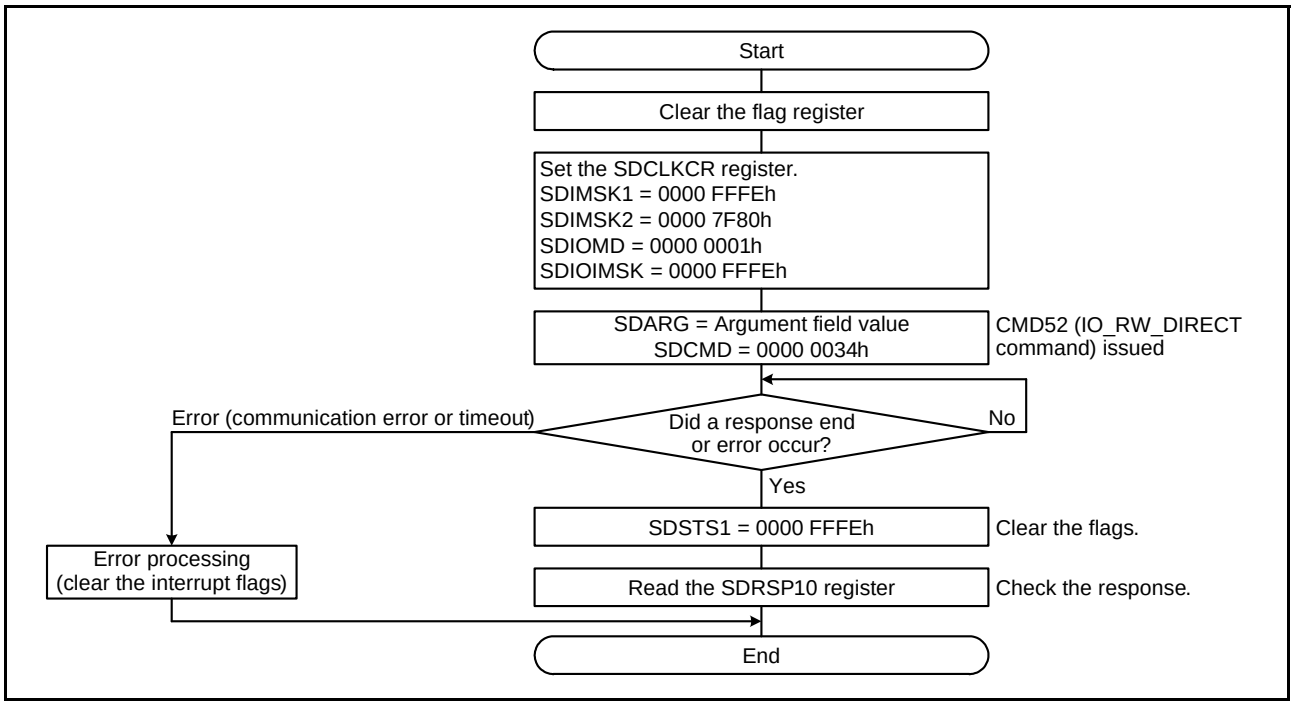



**Figure 40.14 Issuing the IO_RW_DIRECT Command**

### 40.3.6.8 IO_RW_EXTENDED Command (CMD53 (Multi-Block Read))

Figure 40.15 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block read).

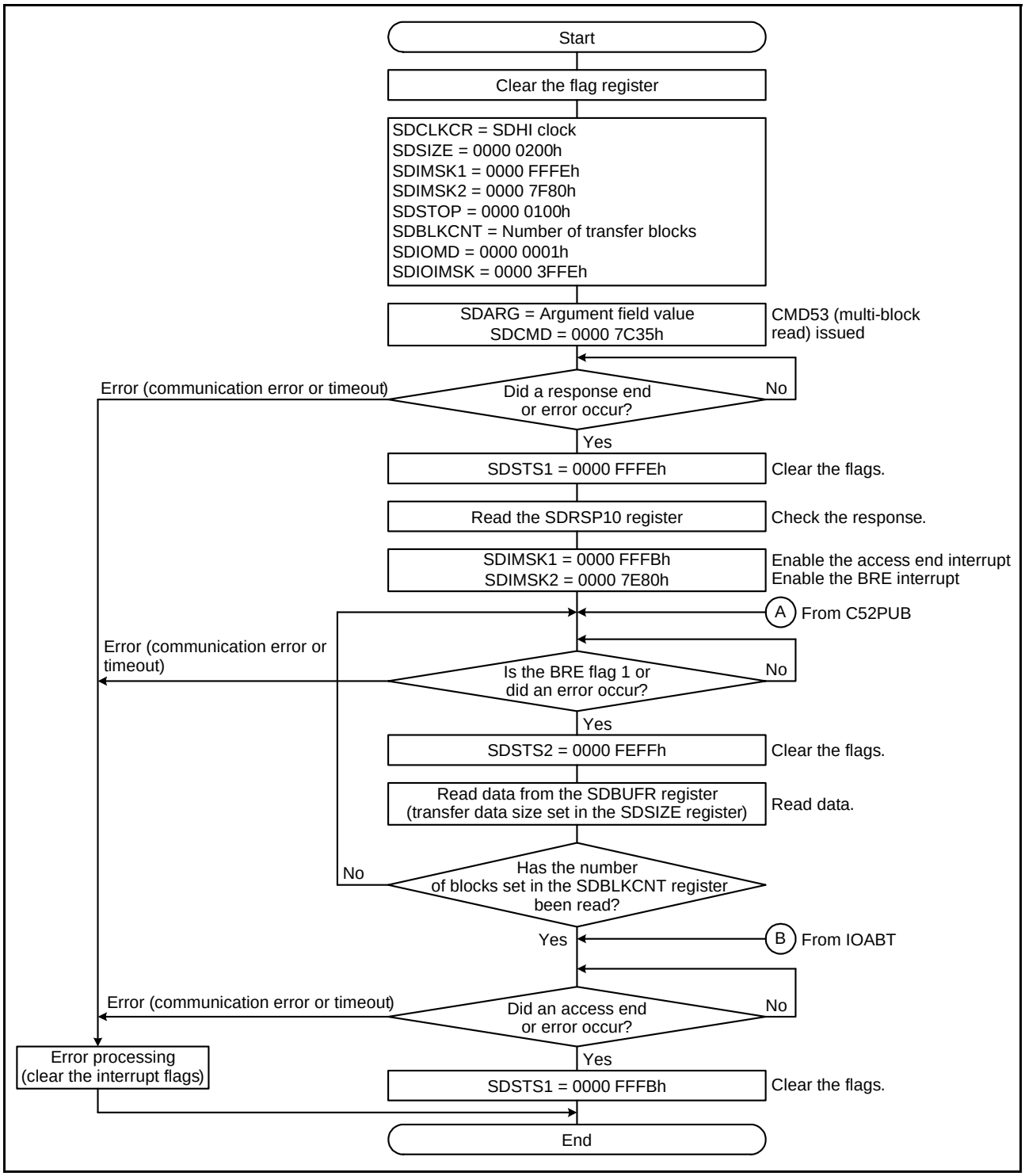



**Figure 40.15 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)**

Figure 40.16 shows an example of entering the read wait state and then issuing the SDIO none abort command (CMD52) during the IO_RW_EXTENDED command (CMD53/Multi-block read).

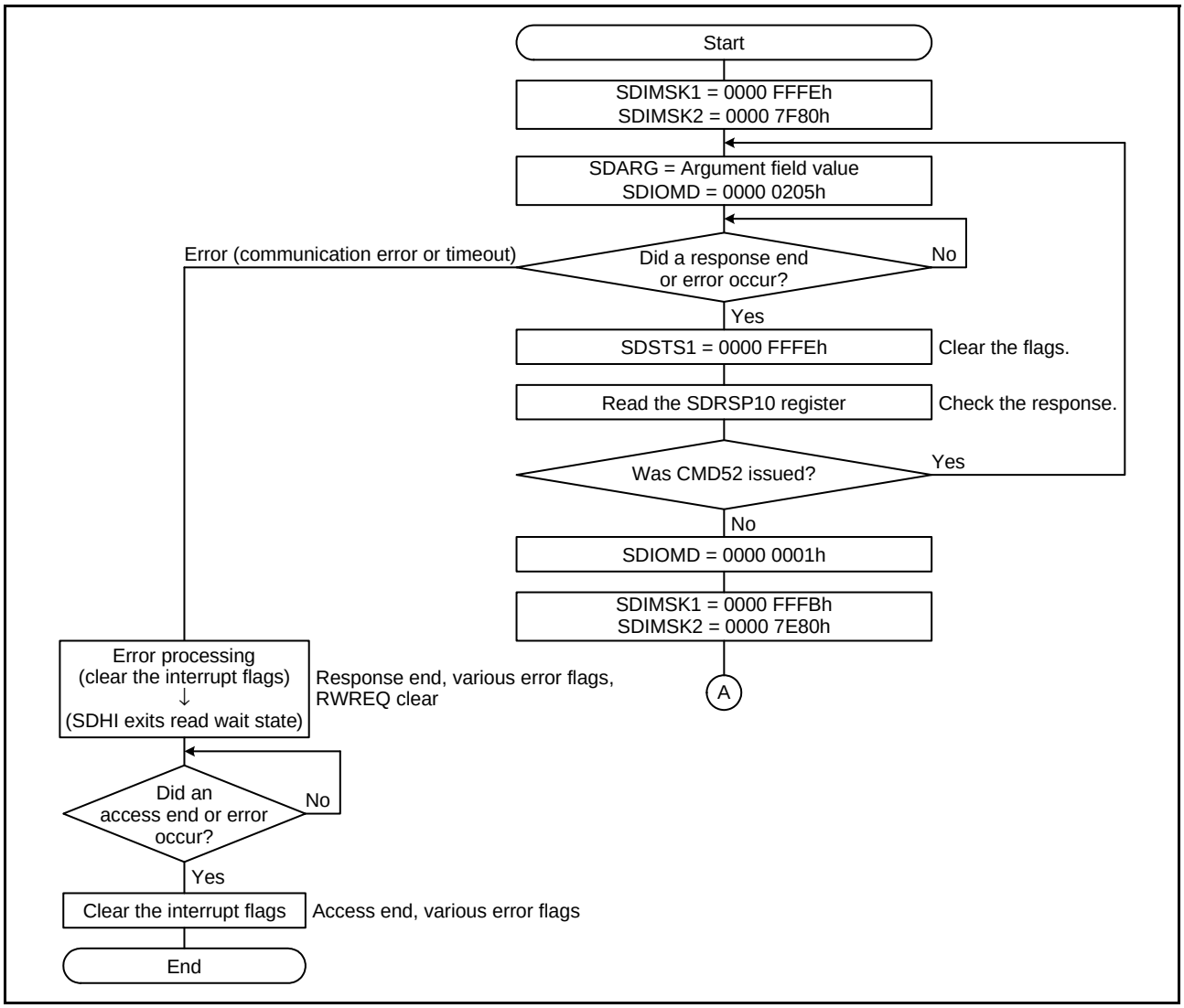



**Figure 40.16 Entering the Read Wait State and Then Issuing the SDIO None Abort Command (CMD52) During the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)**

Figure 40.17 shows an example of SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/ Multi-block read) sequence.

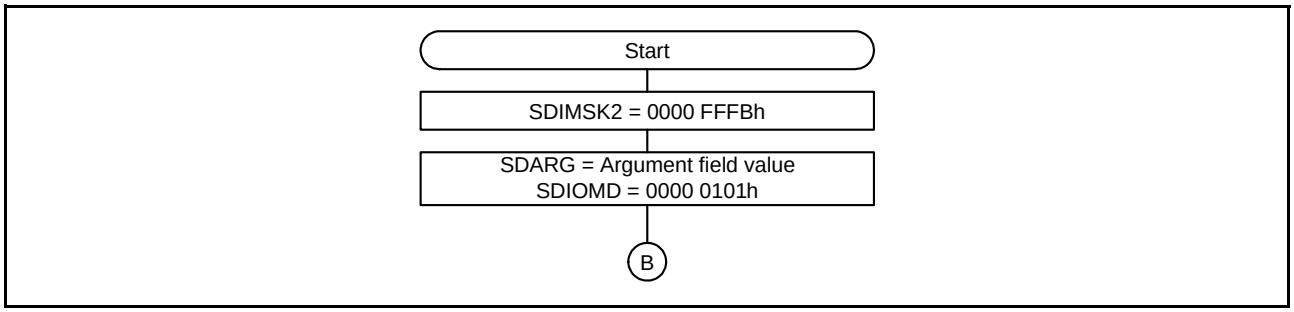



**Figure 40.17 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Read) Sequence**

### 40.3.6.9 IO_RW_EXTENDED (CMD53 Multi-Block Write)

Figure 40.18 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block write).

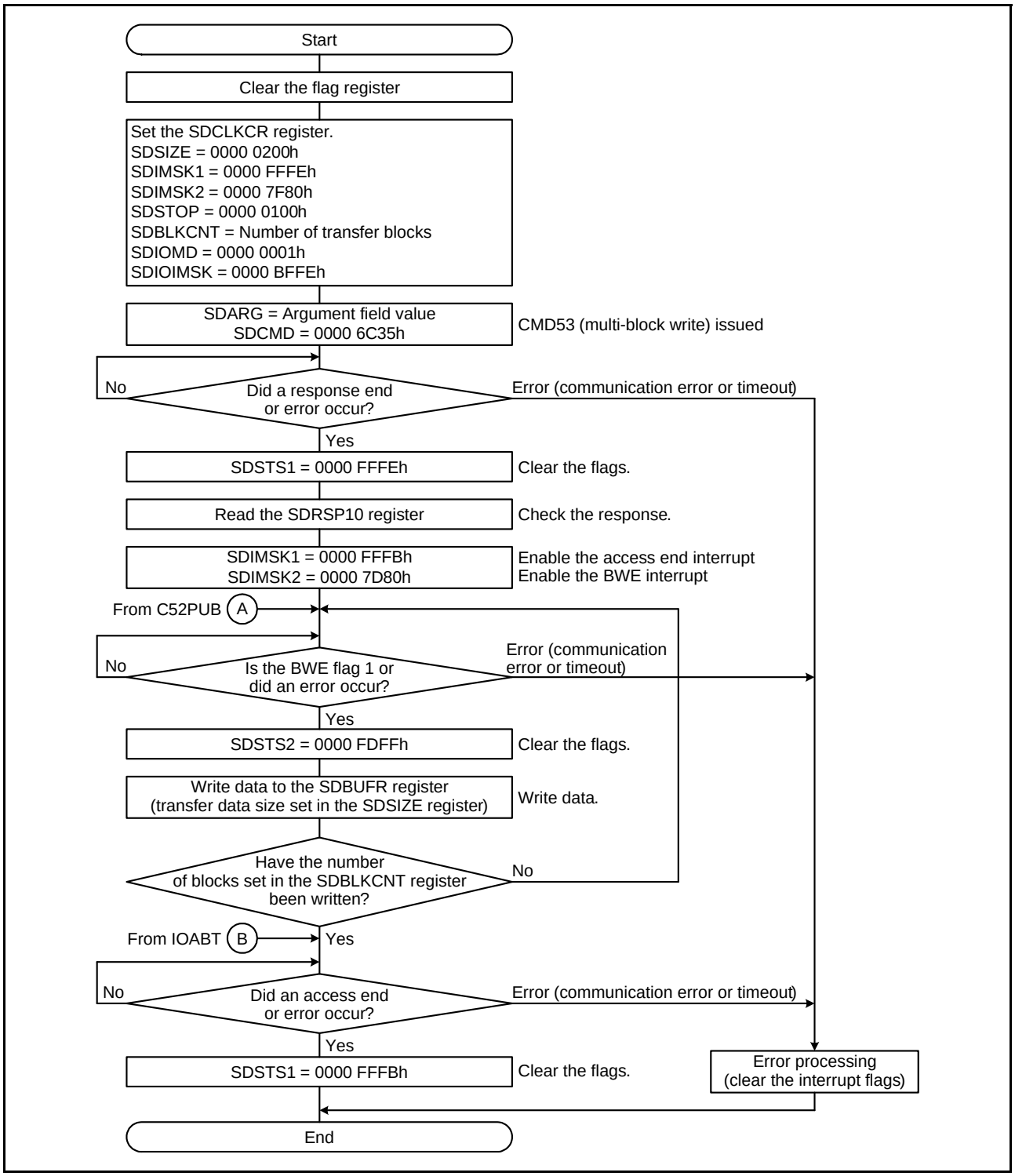



**Figure 40.18 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Write)**

Figure 40.19 shows SDIO none abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/Multi-block write) sequence.

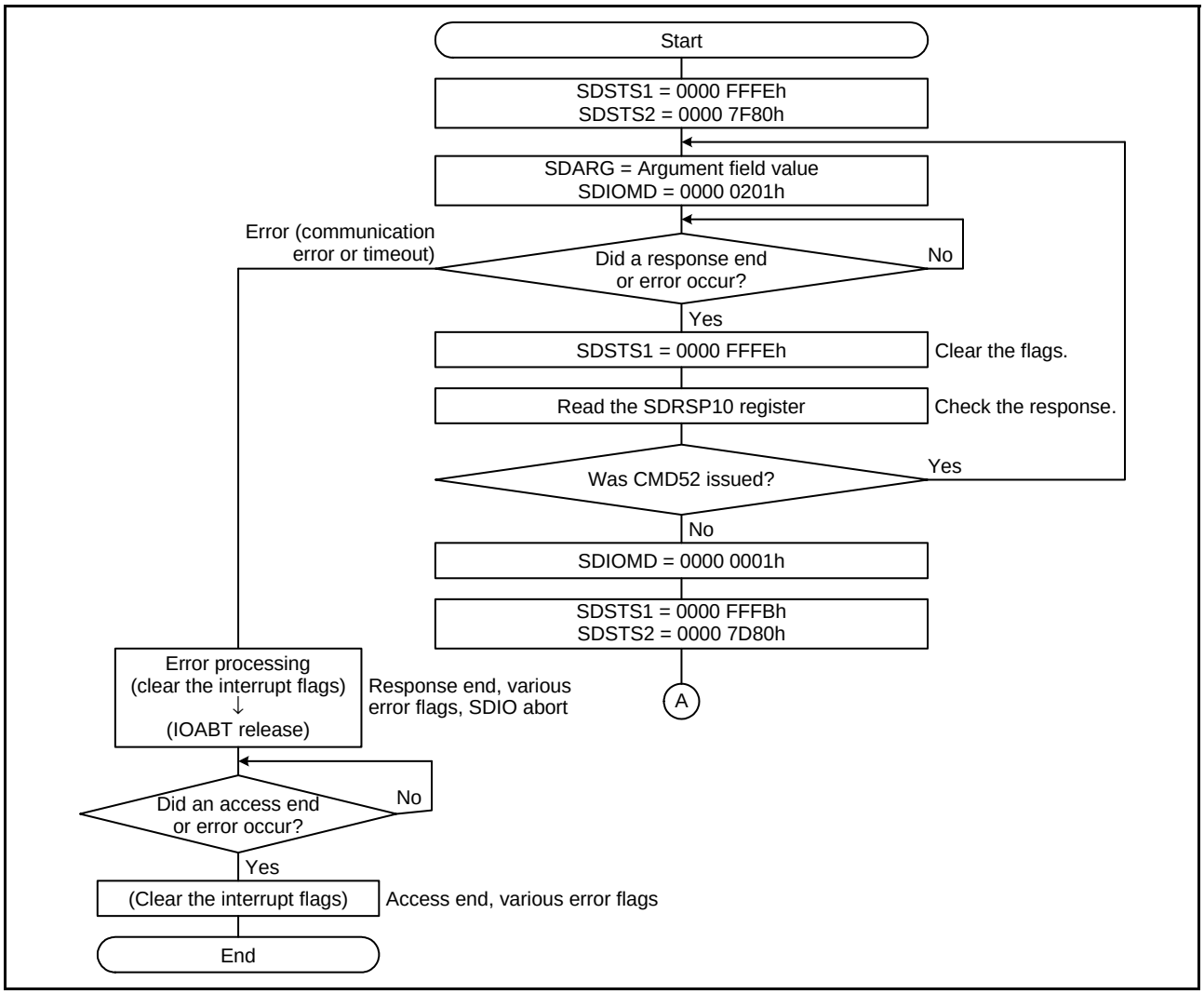



**Figure 40.19 SDIO None Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Write) Sequence**

Figure 40.20 shows SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53) sequence.

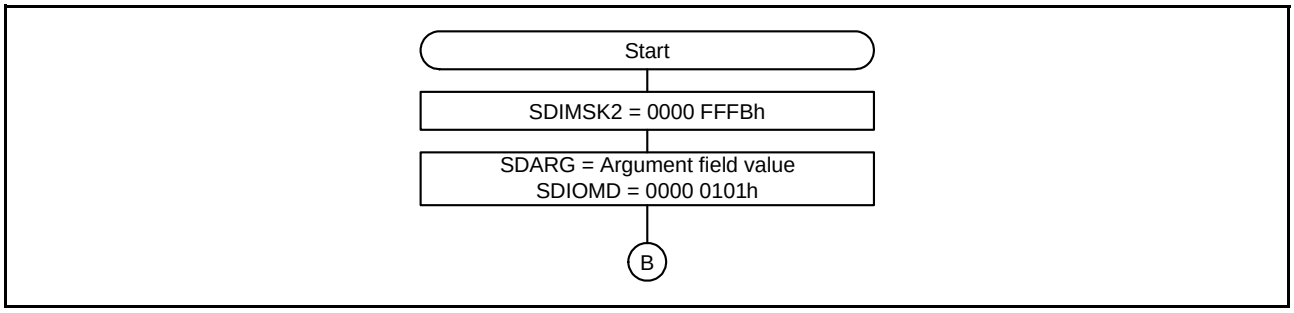



**Figure 40.20 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53) Sequence**

### 40.3.6.10 DMA Transfer

Figure 40.21 shows an example of data being transferred from the SDBUFR register after the CMD18 multi-block read command is issued.

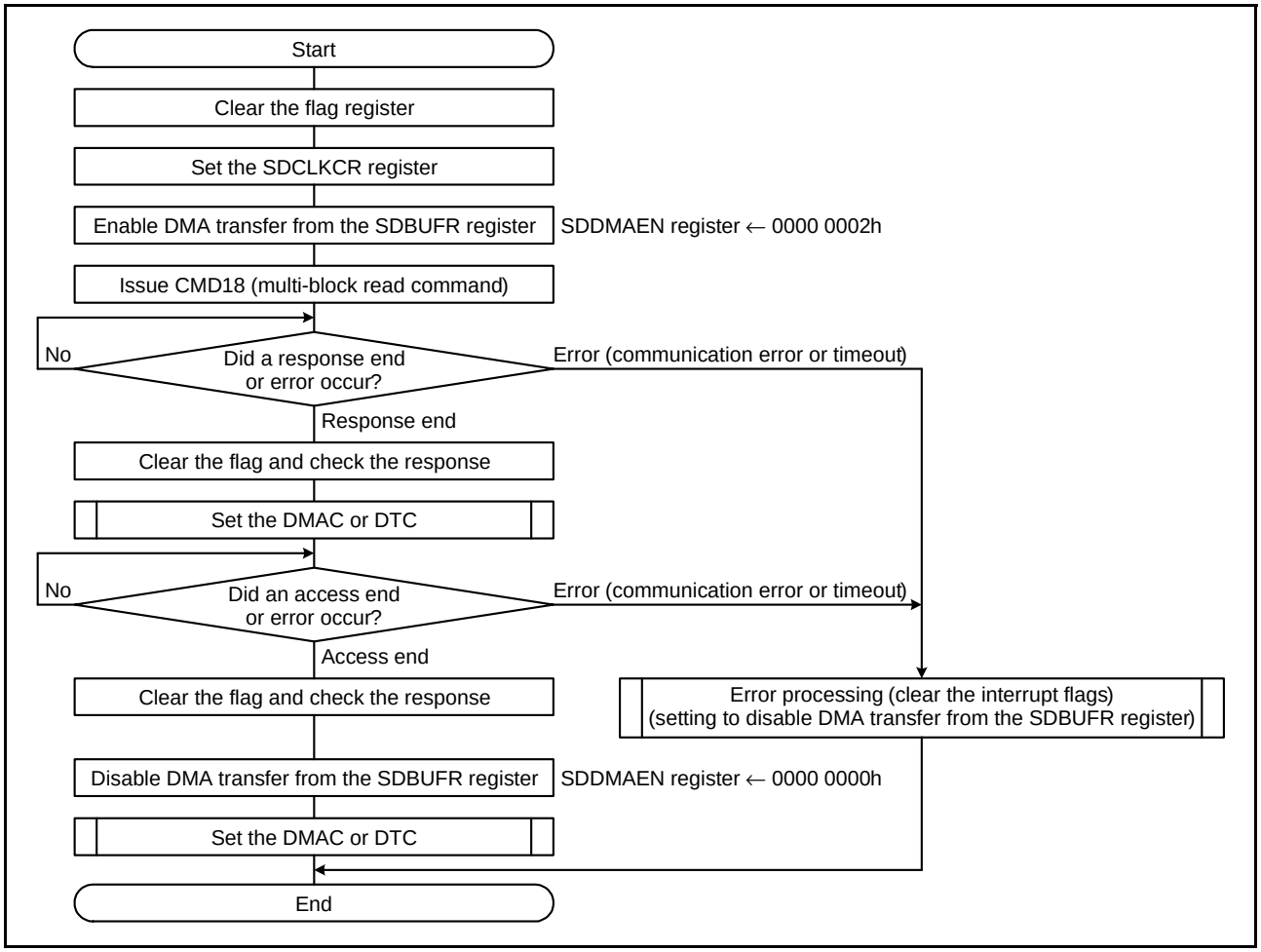



**Figure 40.21 DMA Transfer After CMD18 is Issued**

Figure 40.22 shows an example of data being DMA transferred to the SDBUFR register after the CMD25 multi-block write command is issued.

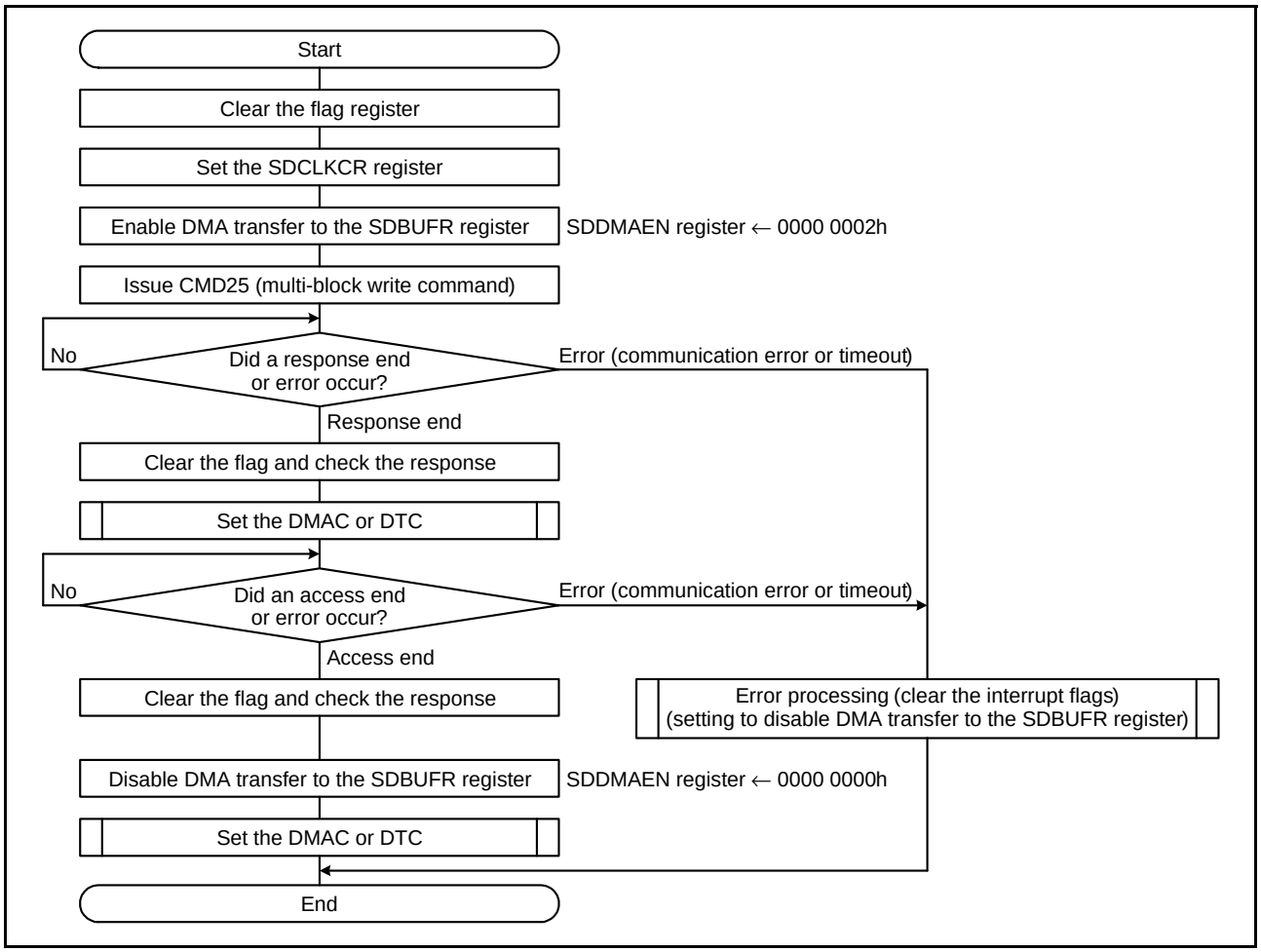



**Figure 40.22 DMA Transfer After CMD25 is Issued**

## 40.4 Interrupts

Table 40.8 lists the SDHI interrupt sources. When the status flags in registers SDSTS1, SDSTS2, and SDIOSTS become 1, if the corresponding bits in registers SDIMSK1, SDIMSK2, and SDIOIMSK are 0, the SDHI requests an interrupt. When clearing the status flags in registers SDSTS1, SDSTS2, and SDIOSTS, write 0 to the status flags to be cleared and write 1 to the states flags not being cleared.

**Table 40.8 Interrupt Sources**

| Interrupt Source | Status Flag Register | | Interrupt Mask/Enable Register | | DMAC/DTC Triggerable |
|---|---|---|---|---|---|
| | Register symbol | Bit symbol | Register symbol | Bit symbol | |
| CACI | SDSTS1 | ACEND | SDIMSK1 | ACENDM | No |
| | | RSPEND | | RSPENDM | |
| | SDSTS2 | ILA | SDIMSK2 | ILAM | |
| | | BWE | | BWEM | |
| | | BRE | | BREM | |
| | | RSPTO | | RSPTOM | |
| | | ILR | | ILRM | |
| | | ILW | | ILWM | |
| | | DTO | | DTTOM | |
| | | ENDE | | ENDEM | |
| | | CRCE | | CRCEM | |
| | | CMDE | | CMDEM | |
| SDACI | SDIOSTS | EXWT | SDIOIMSK | EXWTM | No |
| | | EXPUB52 | | EXPUB52M | |
| | | IOIRQ | | IOIRQM | |
| CDETI | SDSTS1 | SDD3IN | SDIMSK1 | SDD3INM | No |
| | | SDD3RM | | SDD3RMM | |
| | | SDCDIN | | SDCDINM | |
| | | SDCDRM | | SDCDRMM | |
| SBFAI | SDSTS2 | BWE | SDDMAEN | DMAEN | Yes |
| | | BRE | | | |

### 40.4.1 DMA Transfer Triggered by Interrupt Requests

When the SBFAI interrupt is requested, DMA/DTC transfer can be used to write to or read the SDBUFR register. When using the SBFAI interrupt, set the SDDMAEN.DMAEN bit to 1, the SDIMSK2.BWEM bit to 1, and SDIMSK2.BREM bit to 1.

When the SDDMAEN.DMAEN bit is 1, if a write command is issued, the SDSTS2.BWE flag becomes 1; if a read command is issued, the SDSTS2.BRE flag becomes 1. At this point, the SBFAI interrupt request is output. When the last data of a block is transferred (one block is the transfer data size set in the SDSIZE.LEN[9:0] bits), the SBFAI interrupt request is cleared, and the SDSTS2.BWE flag or the SDSTS2.BRE flag becomes 0.

The SBFAI interrupt request is also cleared by following:

- The SDRST.SDRST bit is set to 0 (SDHI software reset).
- The SDSTOP.STP bit is set to 1.
- The SDIOMD.IOABT bit is set to 1.
- The SDDMAEN.DMAEN bit is set to 0.
  However, if the DMAEN bit is set to 1 again before the next command is written to the SDCMD register, the SBFAI interrupt request is output again.

The SBFAI interrupt request will not be cleared when a communication error or timeout occurs during DMA transfer. Perform error processing by software.

The SDSTS2.BWE flag and BRE flag will not become 0 when a communication error or timeout occurs, nor will they become 0 when the SDSTOP.STP bit and SDIOMD.IOABT bit are set to 1. If the SDSTS2.BWE flag or BRE flag remain set to 1, even if a write command or read command is issued, the SBFAI interrupt request will not be output, and the SDSTS2.BWE flag and BRE flag must be set to 0 before issuing the next command.

Table 40.9 lists the DMAC and DTC settings when performing DMA transfer.

**Table 40.9 DMAC and DTC Settings When Performing DMA Transfer**

| Item | | Setting Description |
|---|---|---|
| Transfer mode | | Block transfer mode |
| Transfer data | 1 data | 32 bits |
| | Block size | Size set in the SDSIZE.LEN[9:0] bits divided by 4 |
| Number of block transfers | | Number of transfers set in the SDBLKCNT register |

## 40.5 Notes on Using the SDHI

### 40.5.1 Illegal Read Access During a Multi-Block Read and How To Avoid It

When the multi-block read command (CMD18) is issued to read one or two blocks, if the response to CMD18 stored in the SDRSP10 register is read, the timing of the read access may cause the response to be read incorrectly.
Figure 40.23 shows examples of a normal read and an illegal read when using CMD18 to read two blocks.

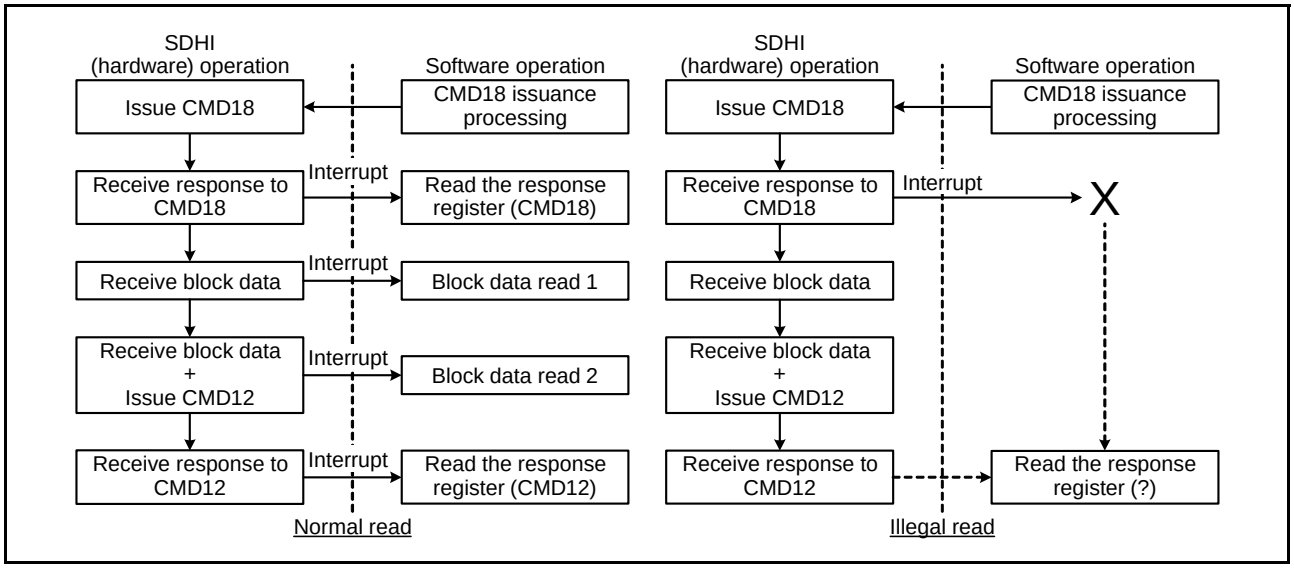



**Figure 40.23 Multi-Block Read Processing When Reading Two Blocks**

In the example of the illegal read, when the interrupt is generated by receiving the response to CMD18, the timing for reading the SDRSP10 register during the interrupt handling is delayed, and instead of the response for CMD18 being read, the response for CMD12 or the data during the response for CMD12 may be read.
This problem can be avoided by performing either of the following:

- When reading one block or two blocks, use a single block read command instead of a multi-block read command.
- When reading the response for CMD18, read the SDRSP54 register instead of the SDRSP10 register.

When using a multi-block read command to read three or more blocks, CMD12 will not be issued if no block data is read, which will prevent this problem from happening. Also, when using a multi-block write command, the above problem can be avoided by transmitting block data after reading the response to CMD25.

### 40.5.2 SDBUFR Register Illegal Write Error

When writing to the SDBUFR register after issuing a single block write command or a multi-block write command, write data for the size set by the SDSIZE.LEN[9:0] bits. If the amount of data written to the SDBUFR register is greater than the size set by the SDSIZE.LEN[9:0] bits, an illegal write error occurs in the SDBUFR register, and the SDSTS2.ILW flag becomes 1. However, the padding data included in the data being written to the SDBUFR register is ignored, so this error does not occur. For example, if the data size set by the SDSIZE.LEN[9:0] bits is an odd number, of the data written to the SDBUFR register, there is a remainder of 1 byte or 3 bytes. Although the extra data is written to the register, no error occurs. If the data size set by the SDSIZE.LEN[9:0] bits is an even number, and there is a remainder of 2 bytes, no error occurs even if these 2 bytes are written to the SDBUFR register.
If data written to the SDBUFR register is not transmitted, the SDSTS2.SDCLKCREN flag may remain set to 0. In this case, in order to set the SDSTS2.SDCLKCREN flag to 1, the SDRST.SDRST bit must be set to 0 and then set back to 1.

### 40.5.3 Automatic Control of the SDHI Clock Output

As per the SD card specifications, after MCU power-on, 74 cycles of the SDHI clock must be output from the host to the SD card before the card initialization command (CMD0) can be issued. Therefore, 74 cycles of the SDHI clock should be output from the SDHI to the SD card before enabling automatic control of the SDHI clock output.
When automatic control of the SDHI clock output is enabled, SDHI clock output stops if the command sequence is ended by a communication error or timeout. Therefore, if it is necessary to change the internal status of the SD card even after the command sequence ends, disable automatic control of the SDHI clock output and output the SDHI clock to the SD card.

### 40.5.4 Restrictions on Setting the C52PUB Bit During a Multi-Block Write Sequence

During a CMD53 multi-block write sequence, if the SDIOMD.C52PUB bit is set to 1, the SDHI issues CMD52 after the SD buffer becomes empty. To immediately issue CMD52, perform one of the procedures below to suspend writing to the SD buffer, and set the C52PUB bit to 1.

Procedure to suspend writing to the SD buffer when not performing DMA transfer (interrupt used)

1. Set the SDIMSK2.BWEM bit to 1 to disable the interrupt, and suspend writing to the SDBUFR register.
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, set the SDIMSK2.BWEM bit to 0 to enable the interrupt, and resume writing to the SDBUFR register.

Procedure to suspend writing to the SD buffer when performing DMA transfer

1. Configure settings to perform DMA transfer every [SDSIZE register setting value × *n* blocks], and suspend writing to the SDBUFR register before setting the SDIOMD.C52PUB bit (*n* = 1, 2, ...).
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, resume DMA transfer to the SDBUFR register.

### 40.5.5 Note on Setting the SDCLKCR Register

The SDCLKCR register cannot be written when the SDSTS2.SDCLKCREN flag is 0. Set the SDSTS2.SDCLKCREN flag to 1 before writing to the SDCLKCR register.

### 40.5.6 Writing to the SDSTOP Register During a Multi-Block Read Sequence

When the SDSTOP.SDBLKCNTEN bit is 1 during a multi-block read sequence, if the SDSTOP.STP bit is set to 1 and the command sequence is stopped, the command sequence may not be completed depending on when the SDSTOP.STP bit is set to 1. To avoid this problem, set the SDSTOP.STP bit and the SDSTOP.SDBLKCNTEN bit to 0 simultaneously. Note that at this time, the SDSTOP.SDBLKCNTEN bit should be set to 0 even if the SDSTS2.SDCLKCREN bit is 0.
If the SDSTOP.SDBLKCNTEN bit is not set to 0, the command sequence can be completed by setting the SDRST.SDRST bit to 0.
During a CMD53 multi-block transfer, when stopping data transfer by setting the SDIOMD.IOABT bit to 1, the SDSTOP.SDBLKCNTEN bit should remain set to 1.

### 40.5.7 Controlling Module Operation

SDHI operation is controlled by setting the MSTPCRD.MSTPD19 bit. Setting the MSTPD19 bit to 0 enables the SDHI; setting the MSTPD19 bit to 1 disables the SDHI. The SDHI is disabled after a reset. Registers in the SDHI can be accessed by setting the MSTPD19 bit to 0. Refer to section 11, Low Power Consumption for details.

# 41. Bluetooth Low Energy (BLE)

This MCU has a Bluetooth Low Energy (BLE), which consists of an RF transceiver compliant with Bluetooth 5.0 Low Energy (single mode), a link layer, and an RF transceiver power-supply.
The BLE is controlled by a Bluetooth middleware available from Renesas Electronics Corporation.

## 41.1 Overview

Table 41.1 lists the Specifications of the BLE. Figure 41.1 is a BLE Block Diagram.

**Table 41.1 Specifications of the BLE**

| Item | Description | | |
|---|---|---|---|
| Features | • RF transceiver compliant with the Bluetooth 5.0 Low Energy specification and a link layer<br>• An on-chip matching circuit allows reducing the number of external parts<br>• On-chip BLE-dedicated high-precision low-speed oscillator (32.768 kHz)<br>• Transmission power is selectable as 0 dBm or +4 dBm.<br>• The minimum receiving sensitivity is selectable from among –105 dBm (125 kbps), –100 dBm (500 kbps), –95 dBm (1 Mbps), or –92 dBm (2 Mbps). | | |
| Bluetooth 5.0 functions | Classification | Function | Remark |
| | Device Address | Public or random address | The address can be set as a desired address. |
| | Advertising | Extended or periodic | |
| | | Multiple advertising | Maximum number of sets: 4 |
| | | Advertising or Scan Response Data | Maximum data length = 1650 bytes |
| | Scanning | Passive, active, or periodic | Number of units for concurrent synchronization with periodic advertising = 2 |
| | | Whitelist or periodic advertiser list | Number of units registered in the whitelist: 4<br>Number of units registered in the periodic advertiser list: 4 |
| | Master or slave | Data transmission or reception | Maximum payload length = 251 bytes<br>MoreData function is supported.<br>Master/slave multi-role function is supported. |
| | Other | Bit rates | 125 kbps, 500 kbps, 1 Mbps, 2 Mbps<br>Bit-rate combinations for transmission and reception can be set as desired. |
| | | Frequency hopping | Channel Selection Algorithm #2 |
| | | Encryption circuit for Bluetooth | On-chip Bluetooth-dedicated AES-CCM (128 bits) circuit |
| Other functions | RF transceiver power-supply (DC-to-DC converter, and linear regulator) (Only the linear regulator is available for the 83-pin LGA product.) | | |
| Other features | In addition to the functions listed above, the 83-pin LGA product also has the following features.<br>• Small PCB trace antenna<br>• Bluetooth-dedicated clock oscillator<br>• Certified as compliant with radio-related laws (technical standards, FCC, ISED, and CE) | | |

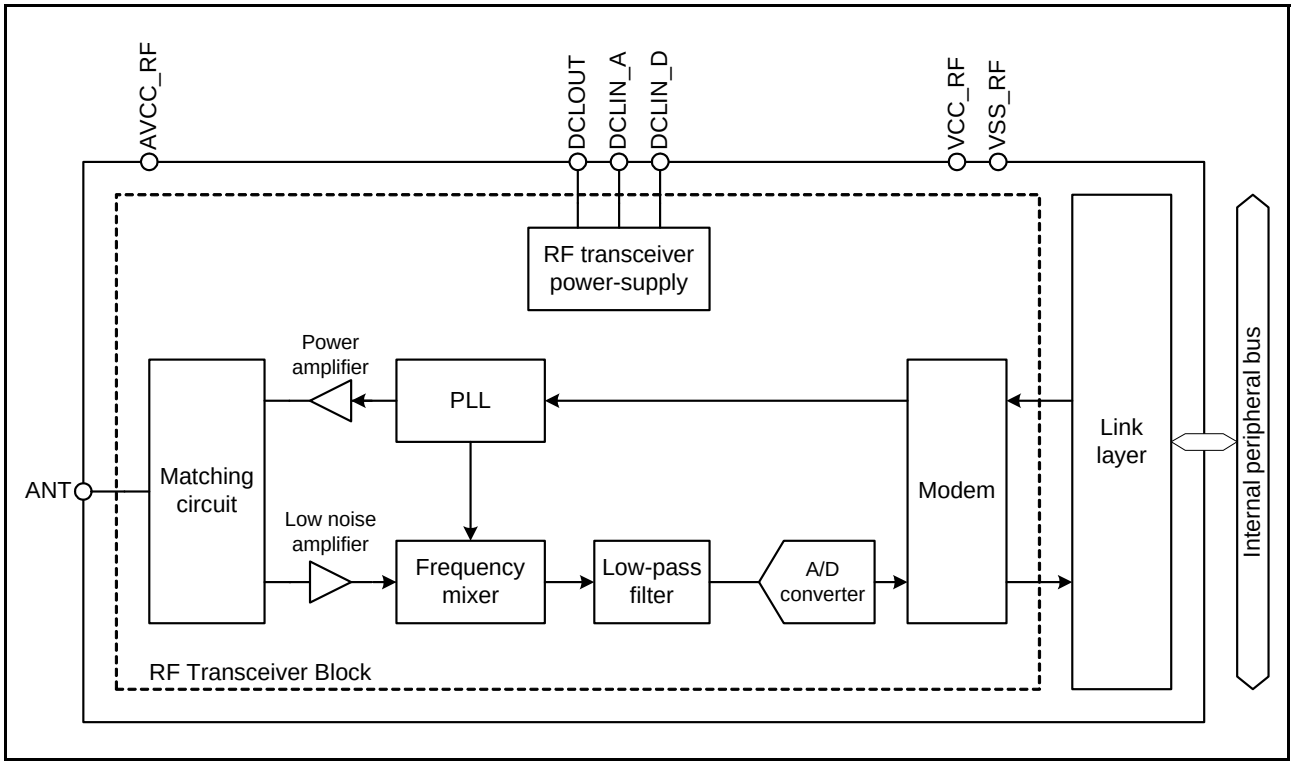



**Figure 41.1 BLE Block Diagram**

Table 41.2 lists the BLE I/O Pins.

**Table 41.2 BLE I/O Pins**

| Pin name | I/O | Description |
|---|---|---|
| ANT | I/O | RF single I/O pin for RF transceiver<br>Set the impedance of the signal line to 50 Ω. |
| DCLOUT | Output | Output pin for the RF transceiver power-supply |
| DCLIN_A,<br>DCLIN_D | Input | If the DC-to-DC converter is to be selected as the power supply for the RF transceiver, connect an inductor and capacitor between the DCLOUT pin and the converter.<br>If the linear regulator is to be selected as the power supply for the RF transceiver, connect a capacitor between the DCLOUT pin and the converter. |
| VCC_RF | Input | RF transceiver power-supply pin |
| AVCC_RF | Input | RF transceiver power-supply pin |
| VSS_RF | Input | RF transceiver ground pin |

Figure 41.2 shows an example of the external connection circuit for the 83-pin LGA version of this MCU.

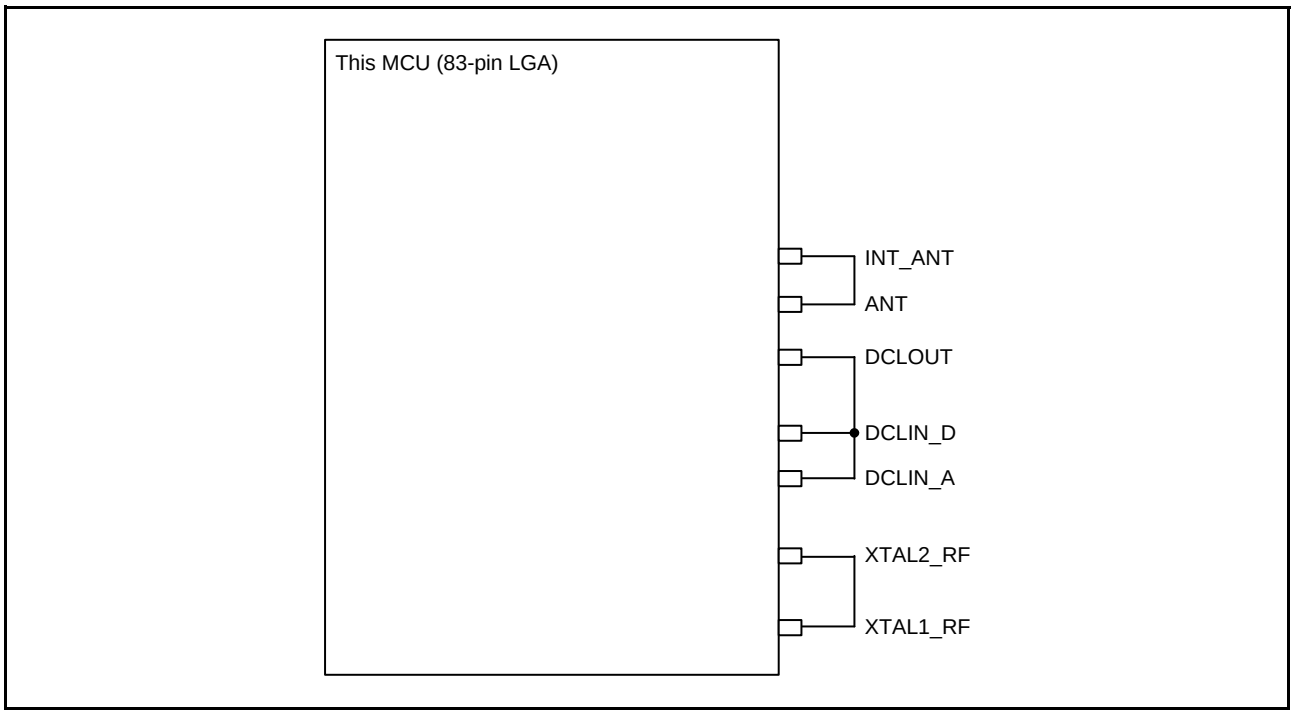



**Figure 41.2 Example of the External Connection Circuit (83-Pin LGA)**

Table 41.3 lists the Bluetooth Low Energy functions that this MCU supports. The use of these functions requires the Bluetooth middleware.

**Table 41.3 List of Supported Bluetooth Low Energy Functions**

| Function | Bluetooth® core spec |
|---|---|
| Low Energy Controller (PHY and LL) | v4.0 |
| Low Energy Host (L2CAP and Security Manager) | |
| Attribute Protocol and Generic Attribute Profile | |
| Appearance Data Type | v4.1 |
| Low Duty Cycle Directed Advertising | |
| 32-bit UUID Support in LE | |
| LE L2CAP Connection Oriented Channel Support | |
| LE Link Layer Topology | |
| LE Ping | |
| LE Data Packet Length Extension | v4.2 |
| LE Secure Connections | |
| Link Layer Privacy | |
| Link Layer Extended Filter Policies | |
| LE 2M PHY | v5.0 |
| LE Coded PHY | |
| High Duty Cycle Non-Connectable Advertising | |
| LE Advertising Extensions | |
| LE Channel Selection Algorithm #2 | |

## 41.2 Operation

### 41.2.1 State Transitions

Figure 41.3 is the State Transition Diagram of the BLE.

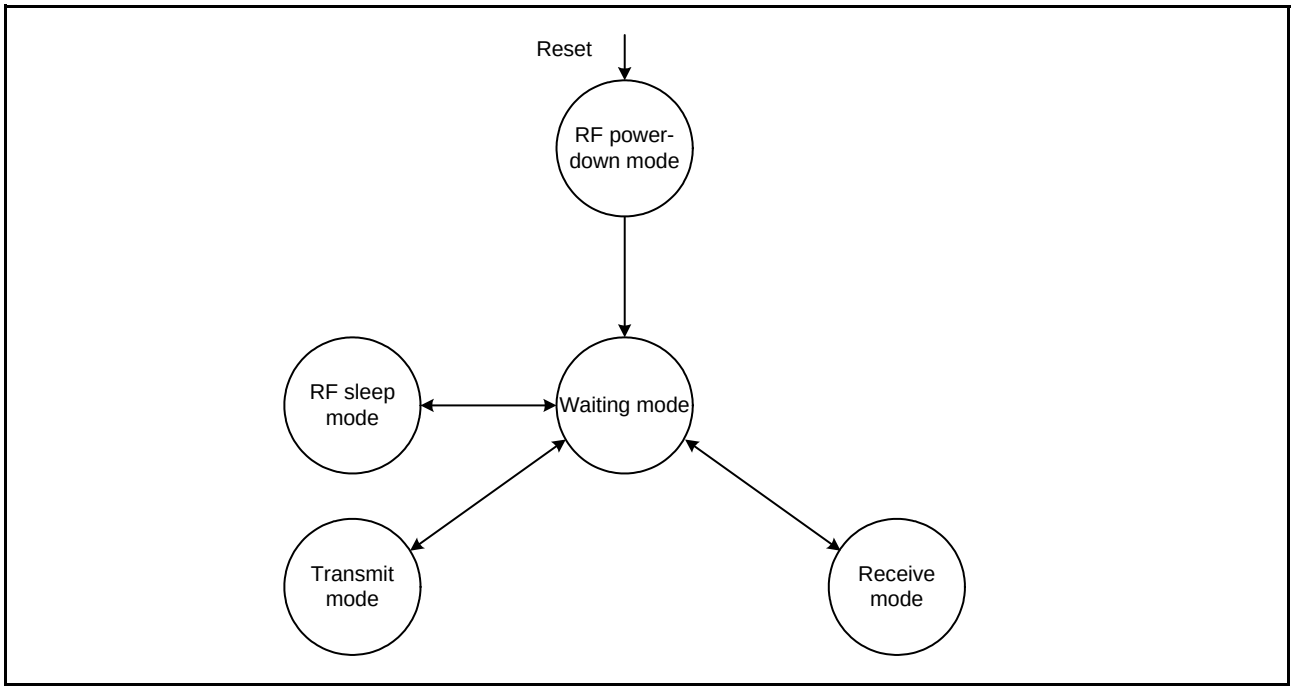



**Figure 41.3 State Transition Diagram of the BLE**

#### RF Power-Down Mode

The RF power-down mode is the initial mode of the BLE after release from the reset state.
Power is supplied to the MCU, but not to the RF transceiver.

#### Waiting Mode

In waiting mode, the BLE waits until the transmission or reception of data or a transition to sleep mode is requested.

#### RF Sleep Mode

The RF sleep mode is a low-power operating mode and the power supply, except to part of the link-layer circuit, is stopped.

#### Transmit Mode

The transmit mode is for the transmission of data.
The mode shifts to waiting mode after the data have been transmitted.

#### Receive Mode

The receive mode is for the reception of data.
The mode shifts to waiting mode after the data have been received.

## 41.3 Interrupts

Table 41.4 shows the interrupt sources.
The Bluetooth middleware executes processing in response to these interrupts.
Do not set the ICU.IERm.IENj bits for these interrupts to 0.

**Table 41.4 BLE Interrupt Sources**

| Name | DTC Activation | DMAC Activation |
|---|---|---|
| BLEIRQ | Not possible | Not possible |
| ERI | Not possible | Not possible |
| RXI | Possible | Possible |
| TXI | Possible | Possible |
| TEI | Not possible | Not possible |

## 41.4 Certificates of Compliance

### 41.4.1 Radio-Related Laws

The 83-pin LGA version of this MCU complies with the radio laws listed below.

- Japan: Certificate of construction type

| Model number | certification number *1 |
|---|---|
| R5F523W8CDLN | 〒 R006-000937 |
| R5F523W8DDLN | 〒 R006-000940 |

- North America: FCC (FCC ID: 2AEMXRX23W8DLN), ISED (IC ID: 20194-RX23W8DLN)*1
- Europe: CE (RE)*2

Note 1. This device is too small to allow indication of the construction type certificate number, FCC ID, or ISED ID on its package.

Note 2. This is based on testing of the radio operation for compliance with the EU Radio Equipment Directive. The user will need to test end products for compliance before affixing the CE mark to them.

### 41.4.2 Bluetooth SIG Certification

QDID: T.B.D.
Name: Renesas Bluetooth low energy Module
Model number: R5F523W8CDLN/R5F523W8DDLN
Product type: End product

### 41.4.3 Labeling and Information RX23W Users Should Provide to End Users of Their Products

Note the following statements when incorporating the 83-pin LGA products into the end products.
This module should be installed in the host product according the installation manual.
The following statements must be described on the user manual of host device of this module;

#### Note to users in the United States of America

#### Caution:

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

#### Declaration of Conformity

This device complies with part 15 of FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

#### Note to users

The following items must be followed when using this certification module:

**List of applicable FCC rules**

This module has been tested for compliance to FCC Part 15

**Summarize the specific operational use conditions**

The moduleis installed in end productformobile RF exposure use condition. Any other usage conditions such as co-location with other transmitter(s) or being used in a portable condition will need a separate reassessment through a class II permissive change application or new certification.

**Limited module procedures**

Any other usage conditions such as co location with other transmitter(s) will need a separate reassessment through a class II permissive change applica tion or new certification.

**Trace antenna designs**

It is strictly forbidden to use antenna except designated.

### RF exposure considerations

This equipment complies with FCC mobile radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with a minimum distance of 20cm between the radiator & your body. If the module is installed in a portable host, a separate SAR evaluation is required to confirm compliance with relevant FCC portable RF exposure rules.

### Antennas

The antenna of this module is PCB type.

This module is certified only with the antenna on the board. No other antenna can be used.

### Label and compliance information

The final end product must be labeled in a visible area with the following: “Contains FCC ID: 2AEMXRX23W8DLN”. The grantee's FCC ID can be used only when all FCC compliance requirements are met.

### Information on test modes and additional testing requirements

This transmitter isinstalled in end product formobile RF exposure condition and any co-located or simultaneous transmission with other transmitter(s) use will require a separate class II permissive change re-evaluation or new certification.

### Additional testing, Part 15 Subpart B disclaimer

This transmitter module is tested as a subsystem and its certification does not cover the FCC Part 15 Subpart B (unintentional radiator) rule requirement applicable to the final host. The final host will still need to be reassessed for compliance to this portion of rule requirements if applicable.

#### Note to users in the United States of America and Canada

#### Note to users

It is strictly forbidden to use antenna except designated.
This equipment must not be co-located or operated in conjunction with any other antenna or transmitter.

#### [For mobile equipment]

This equipment complies with FCC/IC radiation exposure limits set forth for an uncontrolled environment and meets the FCC radio frequency (RF) Exposure Guidelines and RSS-102 of the IC radio frequency (RF) Exposure rules. This equipment has very low levels of RF energy that is deemed to comply without maximum permissive exposure evaluation (MPE).

#### Note to users in Canada

Cet équipement est conforme aux limites d’exposition aux rayonnements énoncées pour un environnement non contrôlé et respecte les règles les radioélectriques (RF) de la FCC lignes directrices d'exposition et d’exposition aux fréquences radioélectriques (RF) CNR-102 de l’IC. Cet équipement émet une énergie RF très faible qui est considérée comme conforme sans évaluation de l’exposition maximale autorisée (MPE).

#### [For portable equipment]

This equipment complies with FCC/IC radiation exposure limits set forth for an uncontrolled environment and meets the FCC radio frequency (RF) Exposure Guidelines and RSS-102 of the IC radio frequency (RF) Exposure rules. This equipment has very low levels of RF energy that is deemed to comply without testing of specific absorption rate (SAR).

#### Note to users in Canada

Cet équipement est conforme aux limites d’exposition aux rayonnements énoncées pour un environnement non contrôlé et respecte les règles les radioélectriques (RF) de la FCC lignes directrices d'exposition et d’exposition aux fréquences radioélectriques (RF) CNR-102 de l’IC. Cet équipement émet une énergie RF très faible qui est considérée comme conforme sans évaluation du débit d’absorption spécifique (DAS).

#### Note to users in Canada

This device complies with Industry Canada’s licence-exempt RSSs. Operation is subject to the following two conditions: (1) this device may not cause interference; and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

**Remarque concermant les utilisateurs au Canada**

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes: (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

**The following information must be indicated on the host device of this module;**

**[for FCC]**

Contains Transmitter Module FCC ID: 2AEMXRX23W8DLN.
This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

**[for ISED]**

Contains IC: 20194-RX23W8DLN
European Community Compliance Statement
**Note:**

Hereby, Renesas Electronics Corporation, declares that this R5F523W8xDLN is in compliance with the essential This

**RE Directive**

requirements and other relevant provisions of Directive 2014/53/EU.

## 41.5 Usage Notes

### 41.5.1 RF Transceiver Power-Supply

In the case of the 85-pin BGA and 56-pin QFN products, the RF transceiver power supply is selectable as either the DC-to-DC converter or linear regulator. Only the linear regulator is available for the 83-pin LGA product.
Figure 41.4 shows an example of the externally connected circuit for BLE when the DC-to-DC converter is selected (only available for the 85-pin BGA and 56-pin QFN products). Figure 41.5 and Figure 41.6 respectively show examples of the externally connected circuit for BLE when the linear regulator is selected for an 85-pin BGA or 56-pin QFN product and for an 83-pin LGA product.

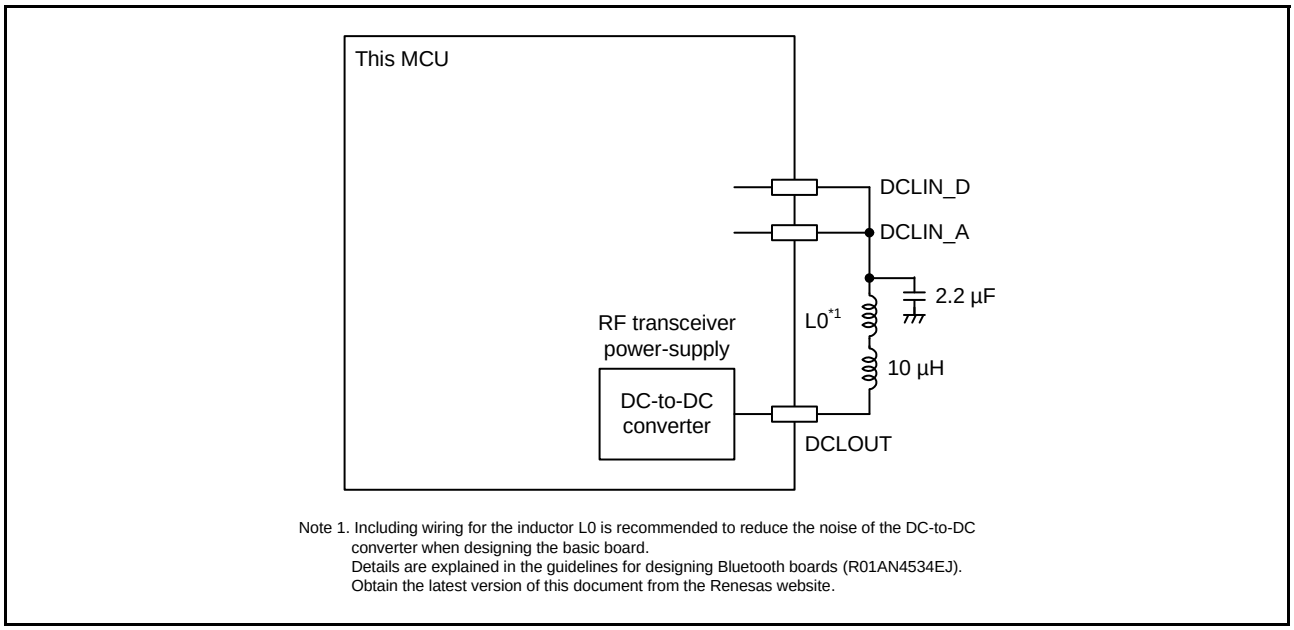



**Figure 41.4 Example of the BLE Externally Connected Circuit When the DC-to-DC Converter is Selected (85-pin BGA, 56-pin QFN)**

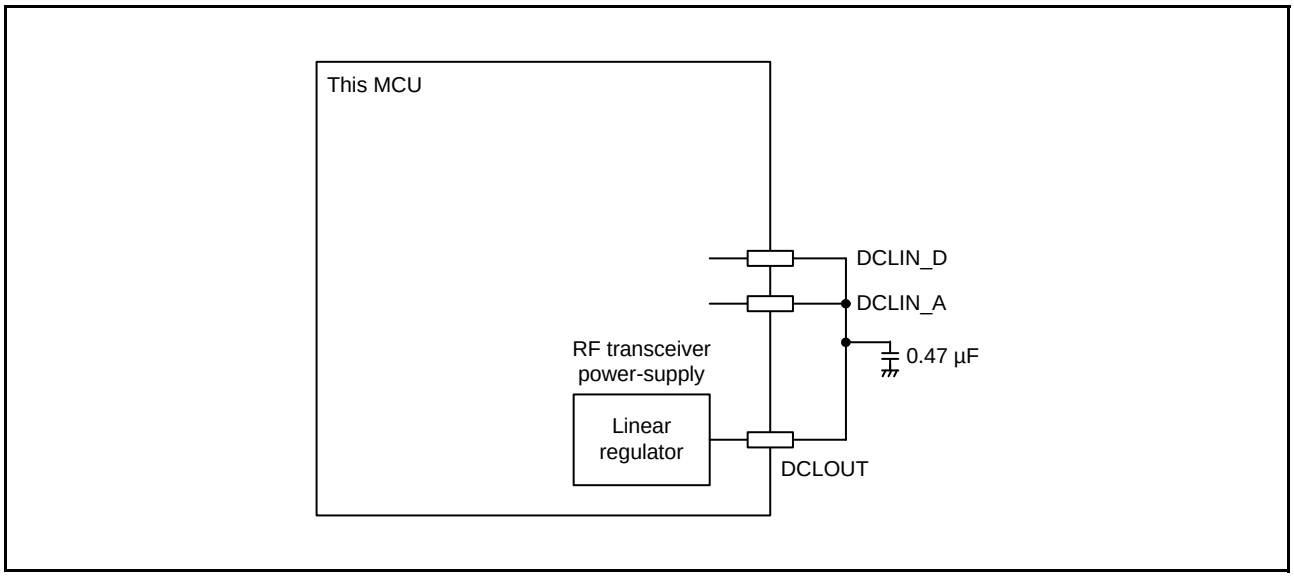



**Figure 41.5 Example of the BLE Externally Connected Circuit When the Linear Regulator is Selected (85-pin BGA, 56-pin QFN)**

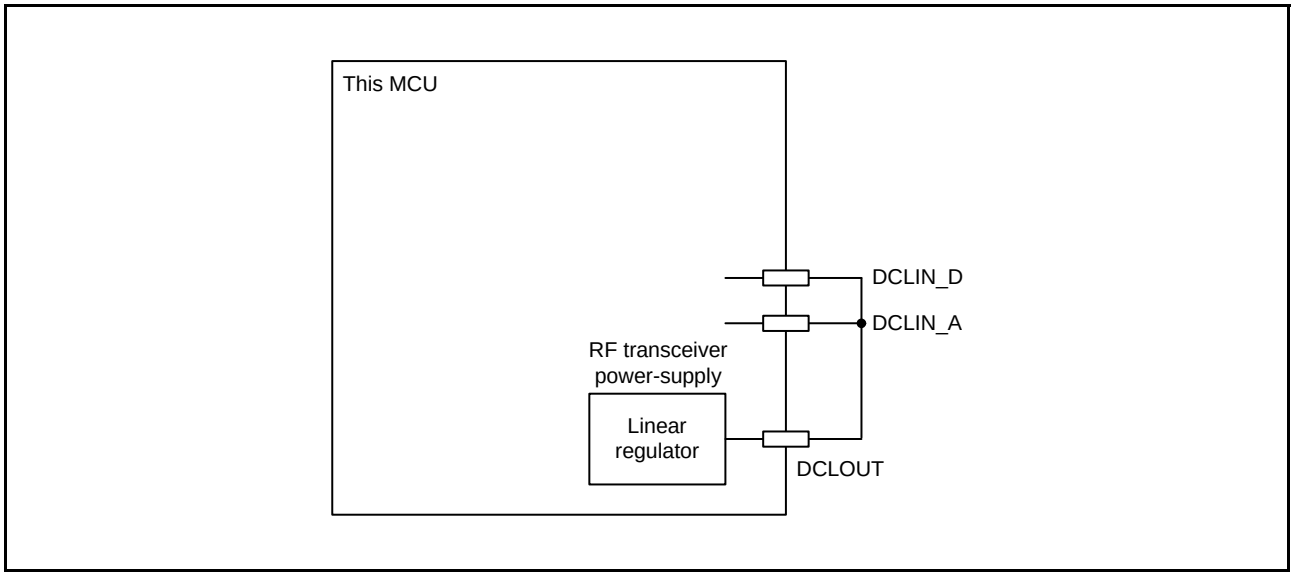



**Figure 41.6 Example of the BLE Externally Connected Circuit (83-pin LGA)**

Do not connect other pins or external circuits to the I/O pins (DCLOUT, DCLIN_D, or DCLIN_A) of the RF transceiver power-supply whether the DC to DC converter or linear regulator is selected.

### 41.5.2 Wireless Standards

International standards and domestic regulations stipulate the use of wireless receivers and transmitters.
Use the device in compliance with the standards or regulations of the country in which the device is being used.

Main standards applicable to the 2.4 GHz band:

- Japan: ARIB STD-T66
- U.S.A.: FCC CFR Title 47 parts 15.247 and 15.249
- Europe: EN 300 328

### 41.5.3 Notes on Board Design

The applicability of the notes on board design differs with the wireless standard to be employed.
Notes on board design are explained in more detail in the guidelines for designing Bluetooth boards (R01AN4534).
Obtain the latest version of this document from the Renesas website.

# 42. Trusted Secure IP (TSIP-Lite)

This MCU incorporates a Trusted Secure IP Lite (TSIP-Lite) module to provide security functions. The module consists of an access management circuit, encryption engine, and random number generator. In combination with the TSIP-Lite library, the TSIP-Lite can prevent eavesdropping (confidentiality), falsification of information (integrity), and impersonation (authenticity).
Key information to be used in encrypting and decrypting data is only stored within the TSIP-Lite, and any external access can be shut out to obtain a system with strong security.

## 42.1 Overview

Table 42.1 summarizes the specifications of the TSIP-Lite. Figure 42.1 shows a block diagram of the TSIP-Lite.

**Table 42.1 Specifications of TSIP-Lite**

| Item | Description |
|---|---|
| Access control | Access management circuit<br>• In case of irregular access to the TSIP-Lite due to a falsified program or runaway execution of a program, this circuit blocks all subsequent access and stops the output of data from the TSIP-Lite. |
| Encryption engine | AES: Compliant with NIST FIPS PUB 197 algorithm<br>• Key sizes: 128 or 256 bits<br>• Block sizes: 128 bits<br>• Block cipher mode of operation<br>ECB, CBC, CTR: Compliant with NIST SP 800-38A<br>CMAC: Compliant with NIST SP 800-38B<br>CCM: Compliant with NIST SP 800-38C<br>GCM: Compliant with NIST SP 800-38D<br>XTS: Compliant with NIST SP 800-38E<br>GCTR<br>• Number of cycles for execution*1<br>ECB, CBC, CTR, CMAC, GCTR, XTS:<br>44 cycles of PCLKB for 128-bit keys, 61 cycles of PCLKB for 256-bit keys<br>CCM:<br>88 cycles of PCLKB for 128-bit keys<br>AES-GCM<br>• AES-GCM is realized by combining AES-GCTR and GHASH.<br>Key management<br>• Keys are only valid within the TSIP-Lite.<br>• Only key generation information is output from the TSIP-Lite.<br>• Keys can be regenerated by the input of key generation information to the TSIP-Lite.<br>Endian<br>• Big or little |
| Generation of random numbers | 32-bit true random number generator<br>• The TSIP-Lite library can assemble 32-bit true random numbers to generate 128- or 256-bit true random numbers.<br>• The generated 128-bit and 256-bit true random numbers are used as keys in encrypting and decrypting data. |
| Protection against illicit key copying | • An ID unique to the MCU (unique ID) is accessible from the access management circuit through the dedicated bus.<br>• Combining the unique ID with the key generation information prevents the illicit copying of the key to another MCU. |
| Supervisor mode | • The supervisor mode signal is connected to the access management circuit and is used to limit control of the TSIP-Lite module to supervisor mode only. |
| Interrupt sources | Three<br>These can be used as triggers for data transfer by the DMAC or DTC. |
| Low power consumption | Setting of the module stop state is possible. |

Note 1. This does not include the overhead for calling functions of the TSIP-Lite library.

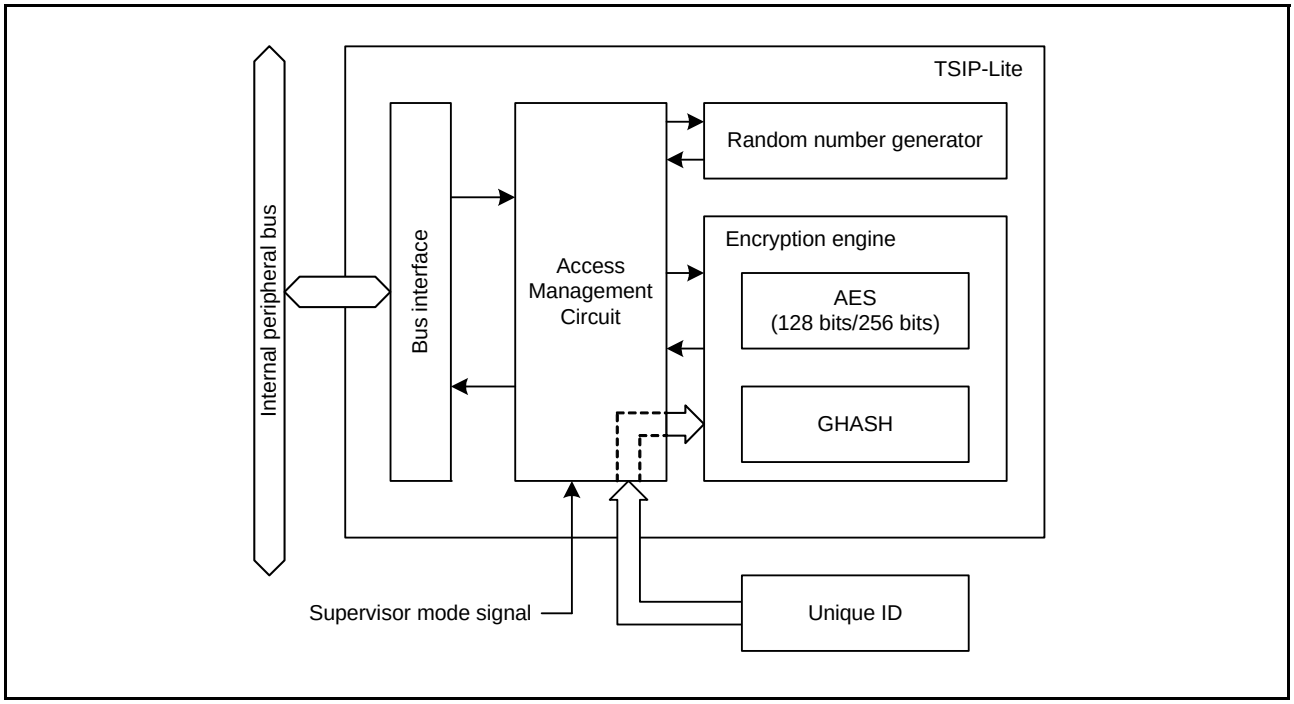



**Figure 42.1 TSIP-Lite Block Diagram**

## 42.2 Operation

### 42.2.1 Operating Modes and State Transitions

Figure 42.2 shows the state transitions of the TSIP-Lite.
Use of the TSIP-Lite security functions is only possible through use of the TSIP-Lite library provided by Renesas Electronics, in accordance with the state transitions as shown in the figure below.
When irregular access to the TSIP-Lite (access that violates the defined procedure) due to a falsified program or a program entering runaway execution, etc. is attempted, the access management circuit does not accept any subsequent access and stops the output of any data from the TSIP-Lite.

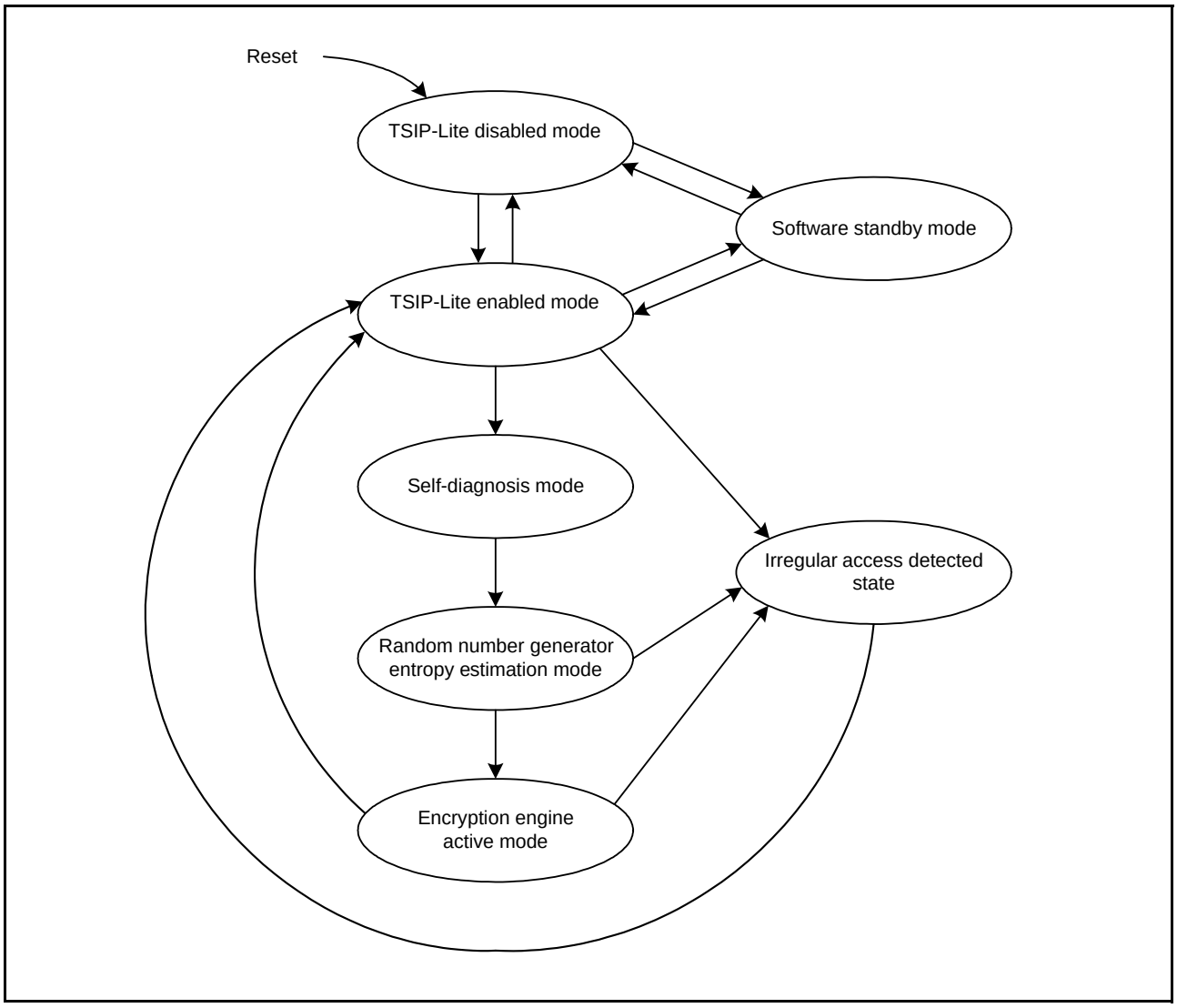



**Figure 42.2 TSIP-Lite Operating Modes and State Transitions**

Many of the security functions that the TSIP-Lite offers are applicable only in the encryption engine active mode. The operations that can be performed in this mode are given below.

(1) Key Installation
(2) Encryption and decryption
(3) Key generation
(4) Random number generation

### 42.2.2 Encryption Engine

Figure 42.3 shows processes of the encryption engine integrated in the TSIP-Lite.
The encryption engine, using the key generation information, performs plaintext to ciphertext encryption and ciphertext to plaintext decryption by hardware.
In no part of the encryption or decryption process, is key data or intermediate data ever exposed outside of the TSIP-Lite.

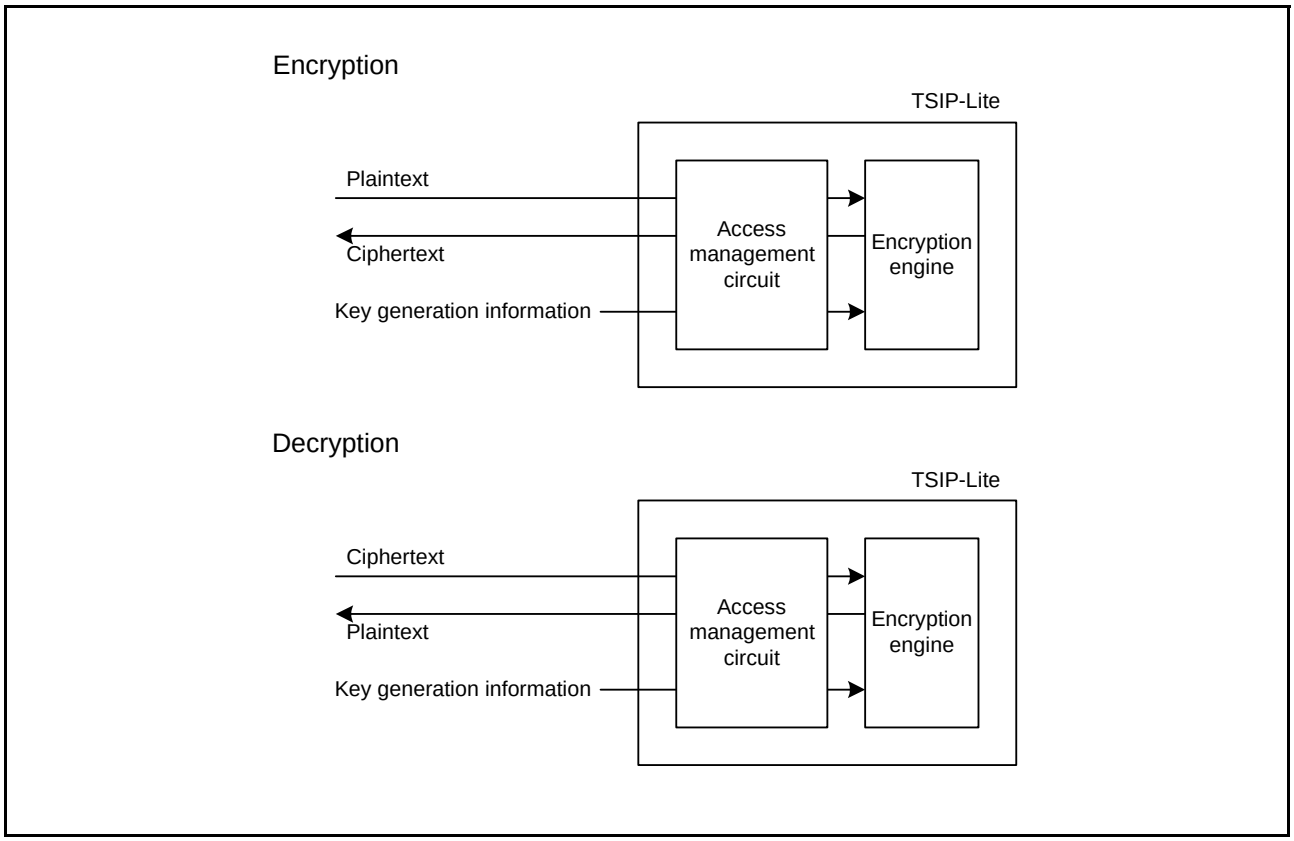



**Figure 42.3 Encryption and Decryption processes by Encryption Engine**

### 42.2.3 Key Installation

The key installation is the operation that safely converts the user key to the key generation information and stores it in flash memory. The procedure for installing the key data are given below.

(1) The user uses the key (Key-2) used for encrypting the user key to encrypt the user key (Key-1) producing eKey-1.
(2) The user sends the encrypted user key (eKey-1) to the TSIP-Lite over the serial interface.
(3) The key generation information of the Key-2 (Index-2) contained in the TSIP-Lite library is used to recover the Key-2, which is then used to decrypt the user key.
(4) The user key is converted to user key generation information (Index-1) using the unique ID and a random number, and stored in flash memory.

The installation process and flow chart are given in Figure 42.4 and Figure 42.5, respectively.
Once the key data is installed, the user key generation information (Index-1) can then be used to perform encryption or decryption.

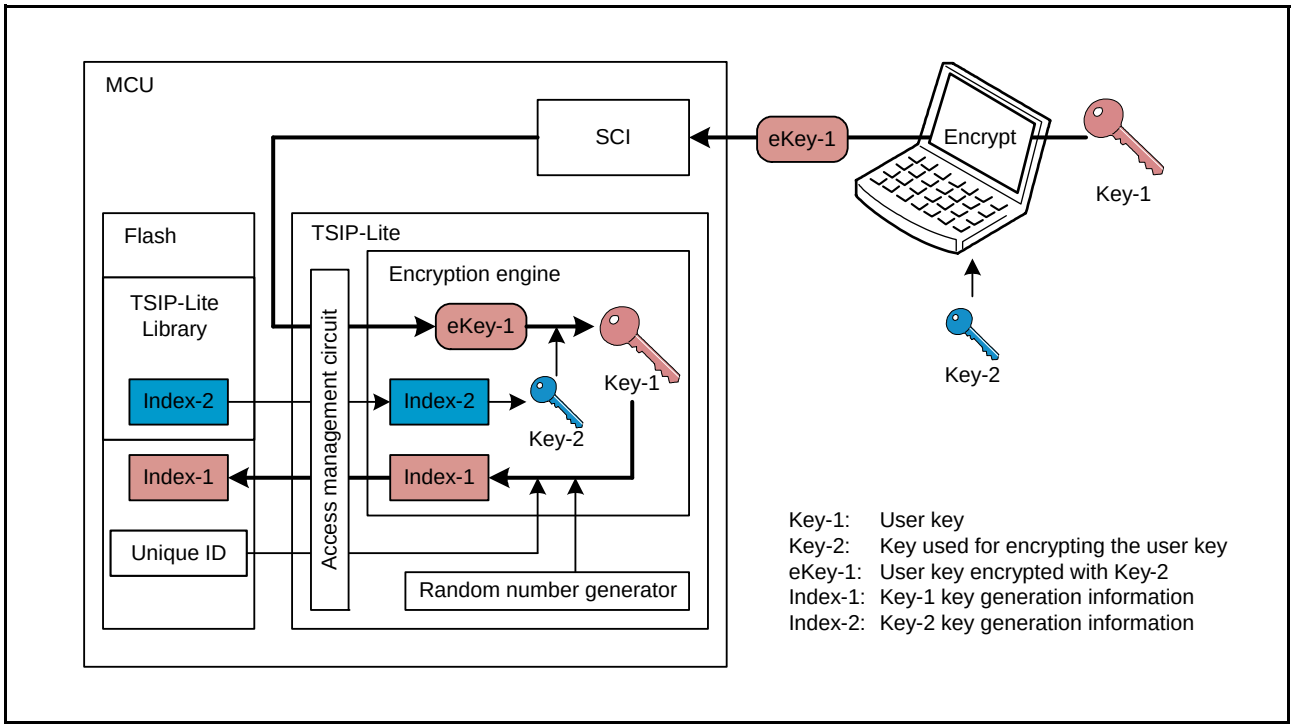



**Figure 42.4 Key Installation Process**

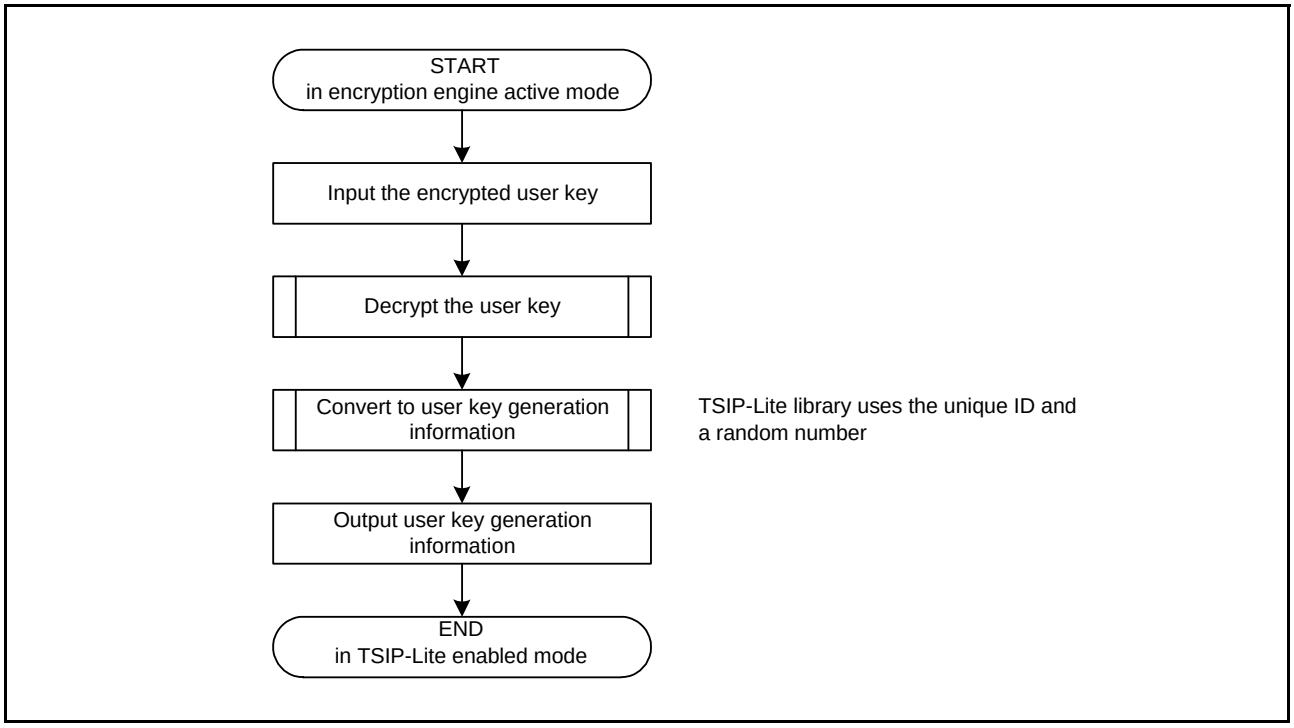



**Figure 42.5 Key Installation Flow Chart**

## 42.2.4 Encryption and Decryption

The procedures for encrypting and decrypting data are given below.

(1) Input the key generation information into the TSIP-Lite, and recover the key data.
(2) Input the data to encrypt or decrypt into the TSIP-Lite. This converts plaintext into ciphertext, and ciphertext into plaintext.
(3) Read the converted data.

The encryption engine has an input buffer and an output buffer, enabling encryption/decryption to proceed in parallel with data input/output.
Figure 42.6, Figure 42.7, and Figure 42.8 show the timing diagram, encryption flow, and decryption flow, respectively.

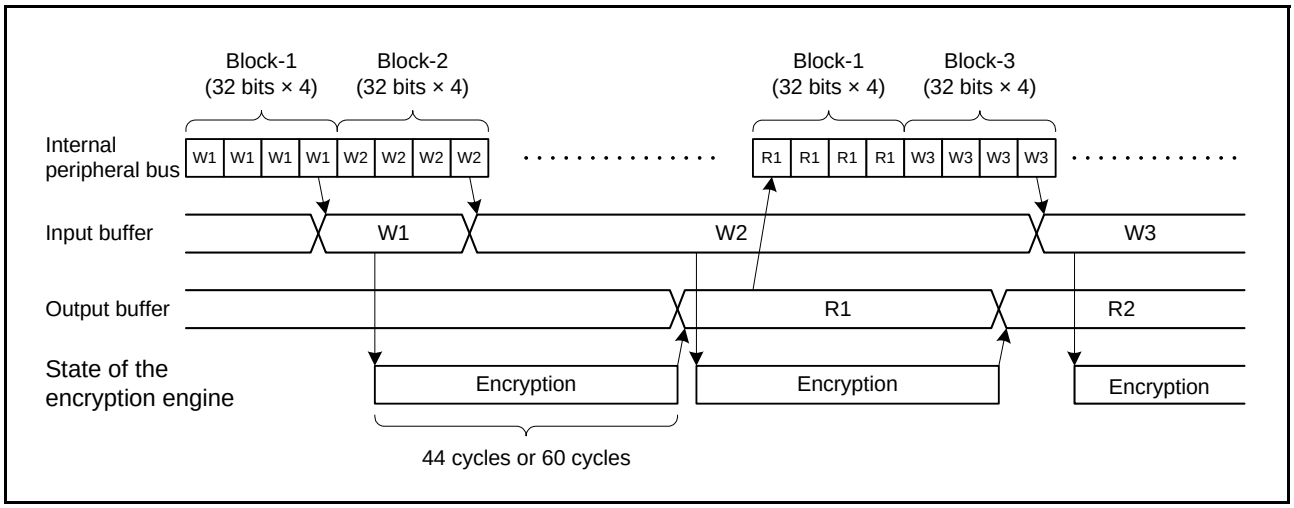



**Figure 42.6 Encryption and Decryption Timing Diagram**

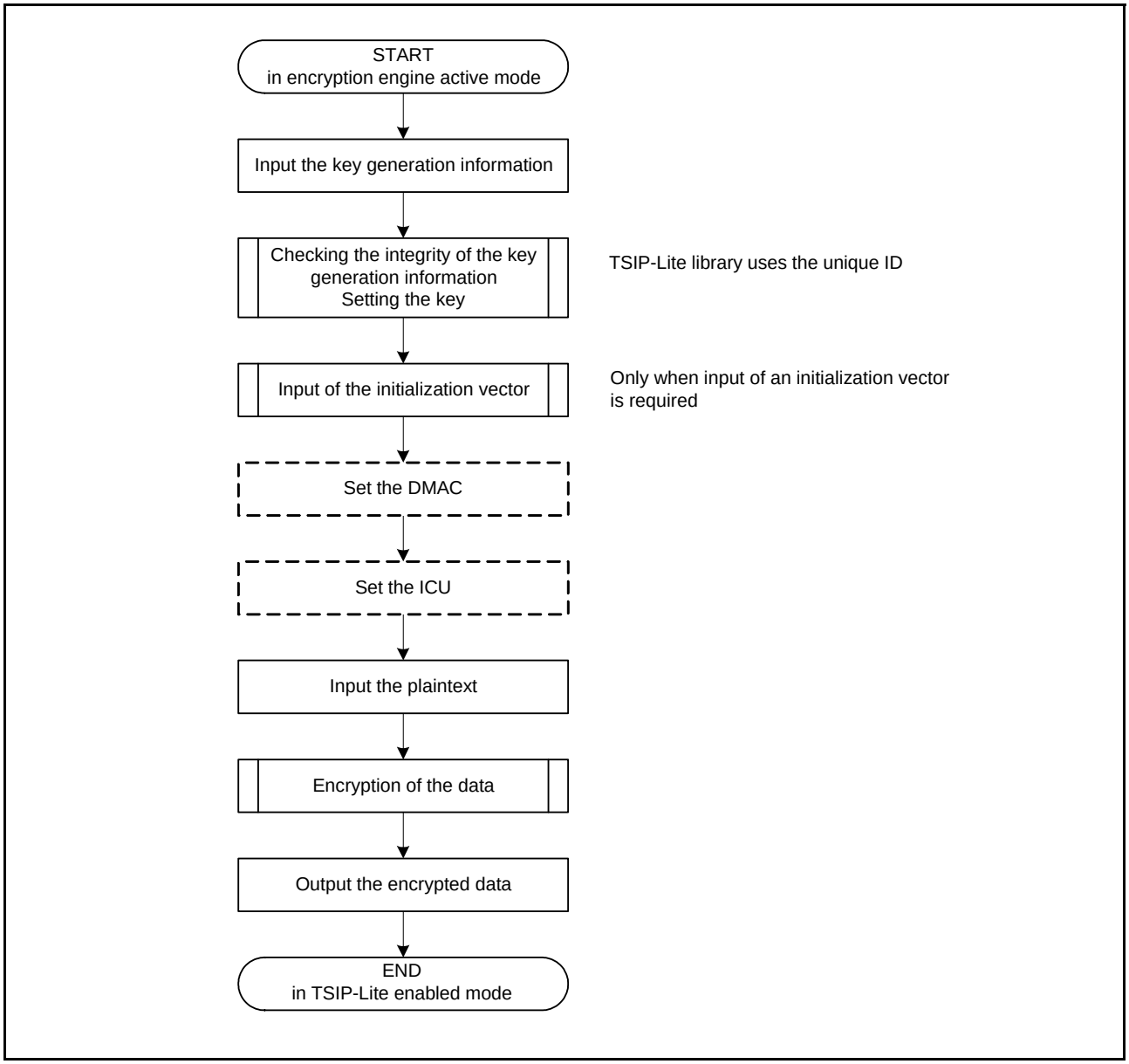



**Figure 42.7 Encryption Flow Chart**

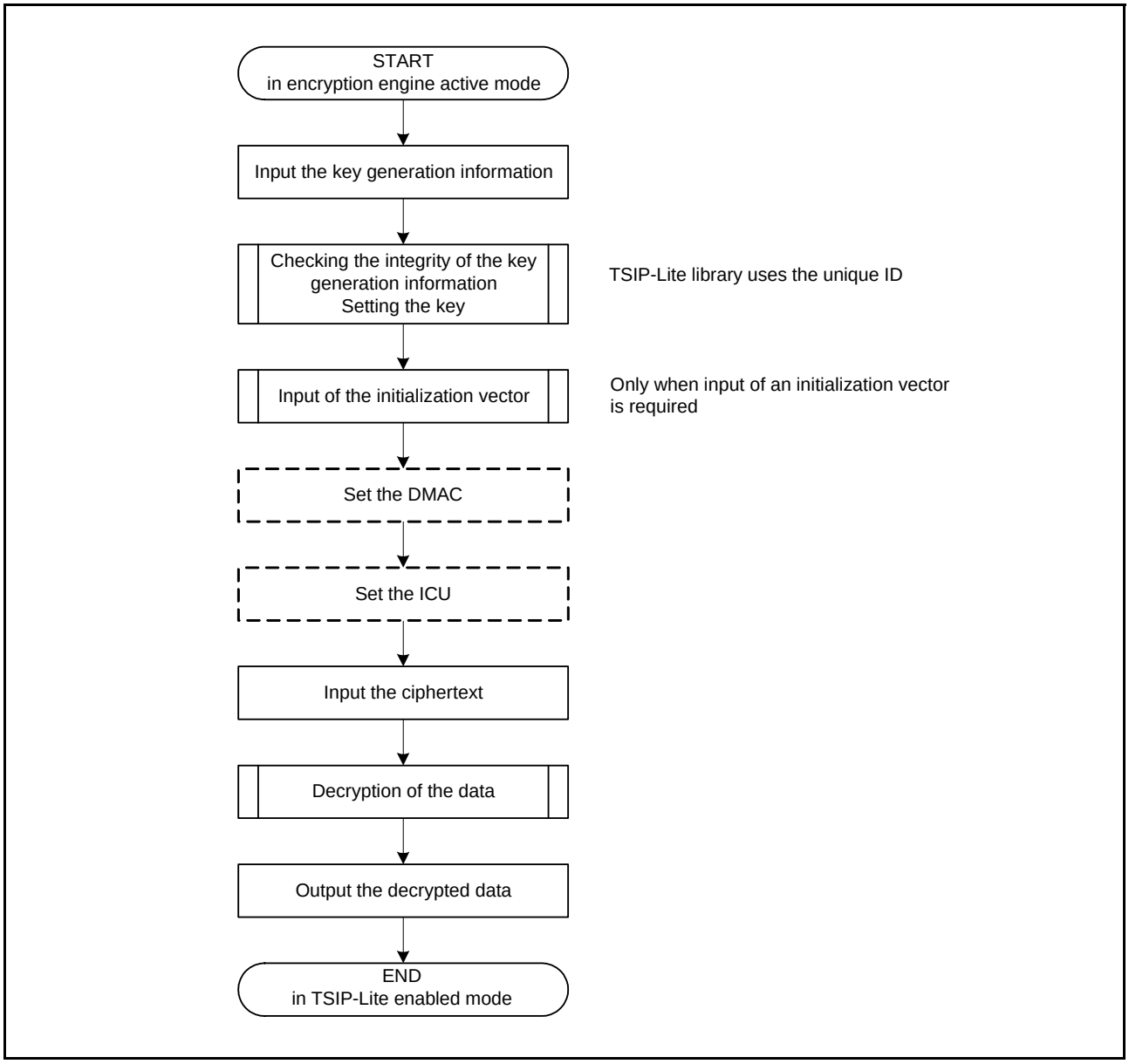



**Figure 42.8 Decryption Flow Chart**

### 42.2.5 Generating Key Generation Information (by Using Random Numbers)

Figure 42.9 shows the generating flow for the key generation information by using random numbers.

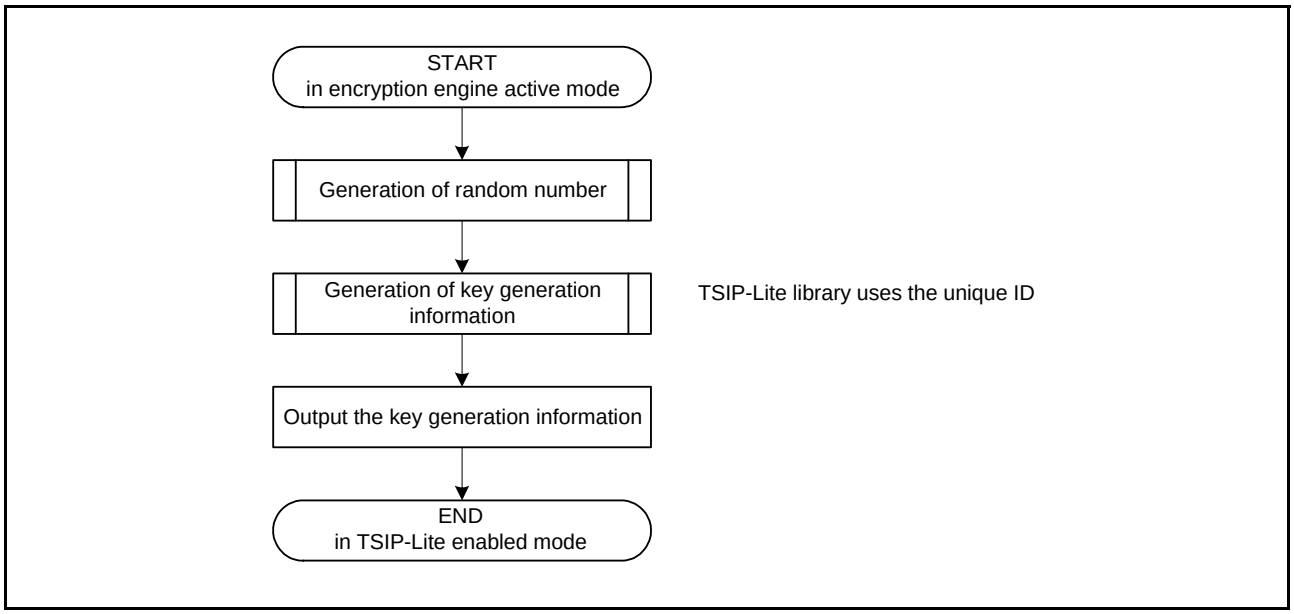



**Figure 42.9 Key Generation Information Generating Flow Chart (Using Random Numbers)**

### 42.2.6 Random Number Generation

Figure 42.10 shows the random number generation flow.

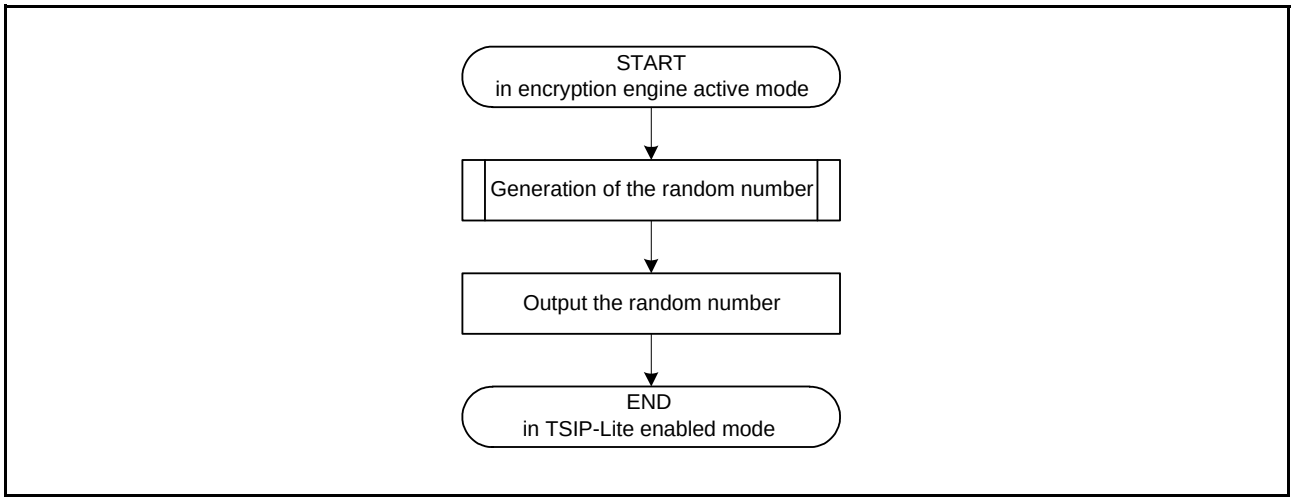



**Figure 42.10 Random Number Generation Flow Chart**

## 42.3 Interrupt

Table 42.2 lists the interrupt sources.
TSIP-Lite library uses interrupts caused by these interrupt sources. Do not set the ICU.IERm.IENj bits corresponding to these interrupt sources to 0.

**Table 42.2 TSIP-Lite Interrupt Sources**

| Name | Interrupt Source | DTC Triggerable | DMAC Triggerable |
|---|---|---|---|
| RD | Data read ready | Yes | Yes |
| WR | Data write ready | Yes | Yes |
| Error | Illegal access detected | No | No |

## 42.4 Usage Notes

### 42.4.1 Standby Mode

When standby mode is entered while the encryption engine is in processing, proper processing cannot be resumed after standby mode is exited. Standby mode should therefore be entered only after first entering TSIP-Lite disabled mode or TSIP-Lite enabled mode.

### 42.4.2 Setting the Module Stop Function

The module stop control register D (MSTPCRD) enables or disables operation of the TSIP-Lite. After a reset, the TSIP-Lite is stopped. After exiting the module stop state, the TSIP-Lite can be accessed. Refer to section 11, Low Power Consumption for details.

### 42.4.3 TSIP-Lite Library

Use of the TSIP-Lite requires the TSIP-Lite library provided by Renesas Electronics. Please contact our sales office for information regarding the TSIP-Lite library.

# 43. Capacitive Touch Sensing Unit (CTSU)

The capacitive touch sensing unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with a dielectric so that a finger does not come into contact with the electrode.

As shown in Figure 43.1, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the value of electrostatic capacitance increases.

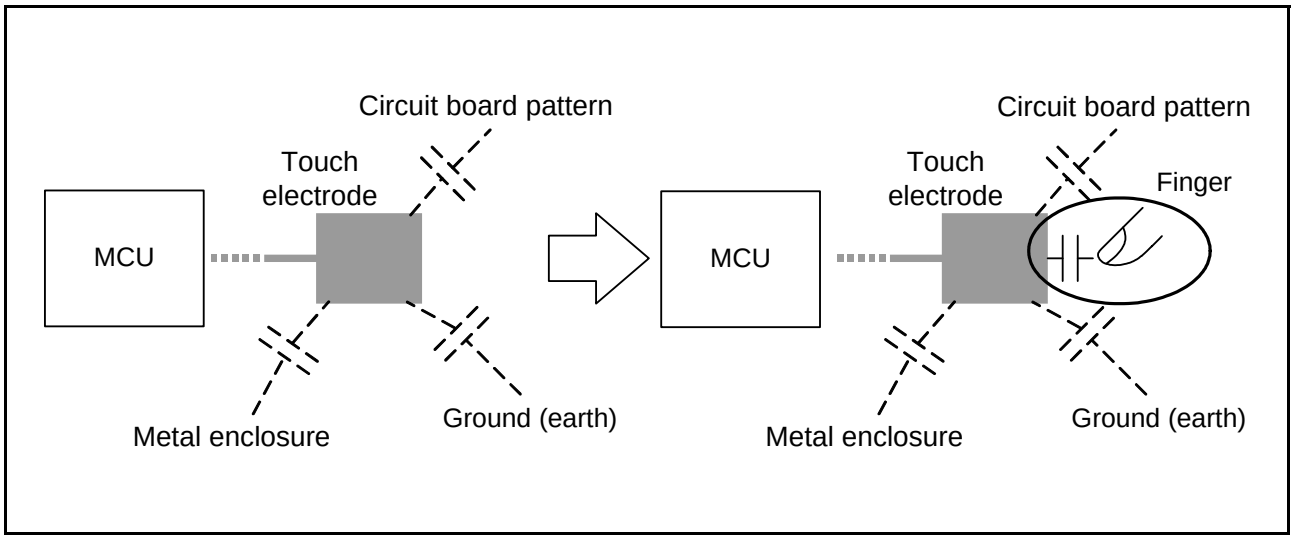



**Figure 43.1 Increased Electrostatic Capacitance Due to Presence of Finger**

Electrostatic capacitance is detected by the following methods: Self-capacitance and mutual capacitance.

In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used as a transmit electrode and a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.

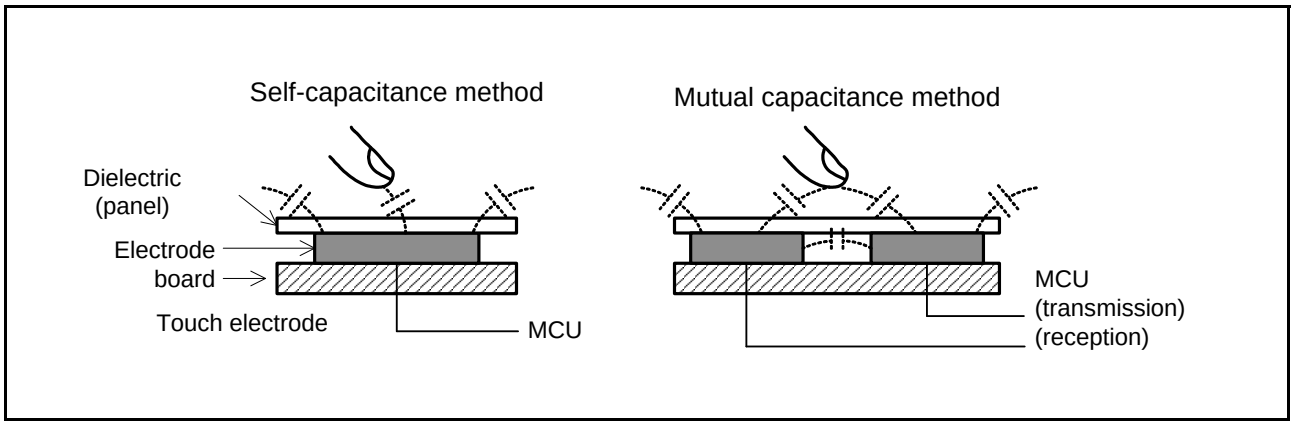



**Figure 43.2 Self-Capacitance Method and Mutual Capacitance Method**

Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged/discharged current, for a specified period.

For details on the measurement principles of the CTSU, refer to section 43.3.1, Principles of Measurement Operation.

In this section, “PCLK” is used to refer to PCLKB.

## 43.1 Overview

Table 43.1 lists the specifications of the CTSU, and Figure 43.3 shows a block diagram of the CTSU.

**Table 43.1 CTSU Specifications**

| Item | | Description |
|---|---|---|
| Operating clock | | PCLK, PCLK/2, or PCLK/4 |
| Pins | TS2, TS3, TS4, TS7, TS8, TS12, TS13, TS22, TS23, TS27, TS30, TS35 | Electrostatic capacitance measurement pins (12 channels) |
| | TSCAP | LPF (low-pass filter) connection pin |
| Measurement modes | Self-capacitance single scan mode | Electrostatic capacitance on a channel is measured by the self-capacitance method. |
| | Self-capacitance multi-scan mode | Electrostatic capacitance on multiple channels is measured successively by the self-capacitance method. |
| | Mutual capacitance full scan mode | Electrostatic capacitance on multiple channels is measured successively by mutual capacitance. |
| Noise prevention | | Synchronous noise prevention, high-pass noise prevention |
| Measurement start conditions | | • Software trigger<br>• External trigger (event input from the event link controller (ELC)) |

As shown in Figure 43.3, the CTSU consists of the status control block, trigger control block, clock control block, channel control block, port control block, sensor drive pulse generator, measurement block, interrupt block, and control registers.

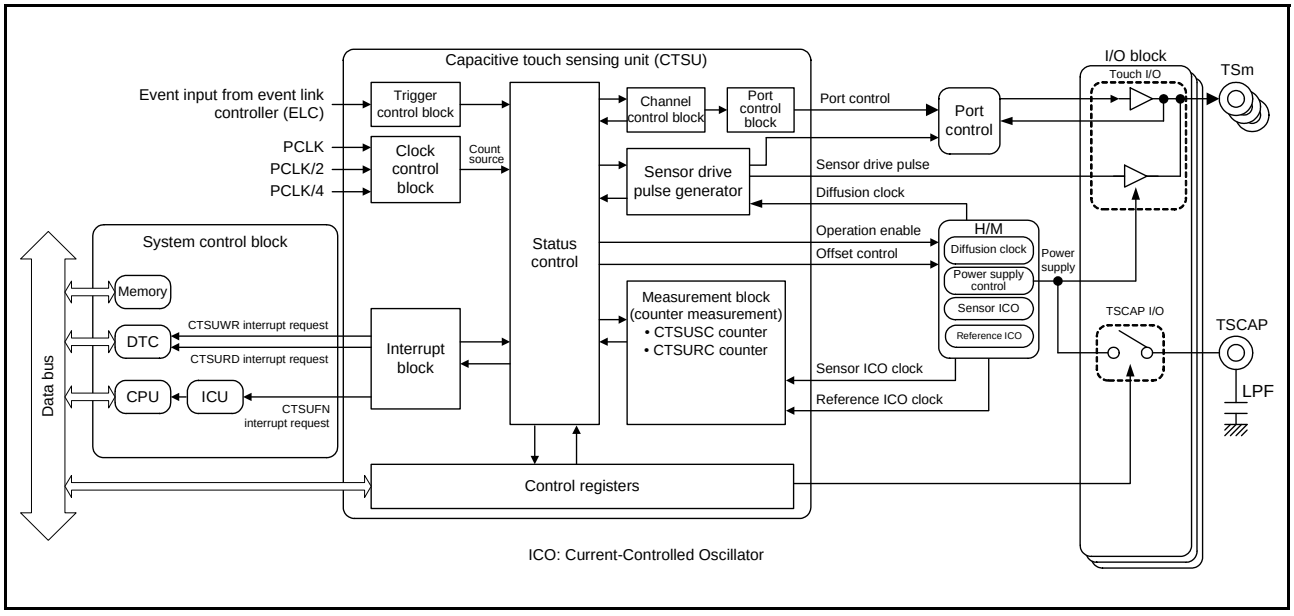



**Figure 43.3 CTSU Block Diagram (m = 2, 3, 4, 7, 8, 12, 13, 22, 23, 27, 30, 35)**

**Table 43.2 CTSU Pin Configuration**

| Pin Name | I/O | Function |
|---|---|---|
| TS2, TS3, TS4, TS7, TS8, TS12, TS13, TS22, TS23, TS27, TS30, TS35 | I/O | Electrostatic capacitive measurement pins (touch pins) |
| TSCAP | — | LPF connection pin |

## 43.2 Register Descriptions

### 43.2.1 CTSU Control Register 0 (CTSUCR0)

Address(es): CTSU.CTSUCR0 000A 0900h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | CTSUINIT | CTSUIOC | CTSUSNZ | CTSUCAP | CTSUSTRT |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CTSUSTRT | CTSU Measurement Operation Start | 0: Measurement operation stops<br>1: Measurement operation starts | R/W |
| b1 | CTSUCAP | CTSU Measurement Operation Start Trigger Select | 0: Software trigger<br>1: External trigger | R/W |
| b2 | CTSUSNZ | CTSU Wait State Power-Saving Enable | 0: Power-saving function during wait state is disabled<br>1: Power-saving function during wait state is enabled | R/W |
| b3 | CTSUIOC | CTSU Transmit Pin Control | 0: The TS pins are driven low<br>1: The TS pins are driven high | R/W |
| b4 | CTSUINIT | CTSU Control Block Initialization | Writing 1 to this bit initializes the CTSU control block and registers*1. This bit is read as 0. | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. The CTSUSC, CTSURC, CTSUMCH0, CTSUMCH1, and CTSUST registers are initialized.

The CTSUCAP and CTSUSNZ bits should be set when the CTSUSTRT bit is 0. These bits can be set at the same time as starting measurement operation.

**CTSUSTRT Bit (CTSU Measurement Operation Start)**

This bit specifies whether CTSU operation starts or stops.
When the CTSUCAP bit is 0 (software trigger), measurement is started by writing 1 to the CTSUSTRT bit, and the CTSUSTRT bit becomes 0 when measurement is finished.
When the CTSUCAP bit is 1 (external trigger), the CTSU waits for an external trigger by writing 1 to the CTSUSTRT bit, and measurement is started at the rising edge of the external trigger. When measurement is finished, the CTSU waits for the next external trigger and operation is continued.
Table 43.3 lists the CTSU states.

**Table 43.3 CTSU States**

| CTSUSTRT Bit | CTSUCAP Bit | CTSU State |
|---|---|---|
| 0 | 0 | Stopped |
| 0 | 1 | Stopped |
| 1 | 0 | During measurement |
| 1 | 1 | During measurement/wait for an external trigger*1 |

Note 1. The state can be read from the CTSUST.CTSUSTC[2:0] flags.
During measurement: CTSUST.CTSUSTC[2:0] flags ≠ 000b
Wait for an external trigger: CTSUST.CTSUSTC[2:0] flags = 000b

If the CTSUSTRT bit is set to 1 when the CTSUSTRT bit is 1, writing is ignored and operation is continued.
To forcibly stop operation (forced stop) when the CTSUSTRT bit is 1, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 simultaneously.

### CTSUCAP Bit (CTSU Measurement Operation Start Trigger Select)

This bit specifies the measurement start condition. For details, see the description of the CTSUSTRT bit.

### CTSUSNZ Bit (CTSU Wait State Power-Saving Enable)

This bit enables or disables power-saving operation during a wait state. This bit can also be used to suspend the CTSU power supply, which decreases power consumption during the wait state.
In the suspended state, the CTSU power supply is turned off while the external TSCAP is still charged after the CTSU power supply has been turned on and the TSCAP has been charged.

**Table 43.4 CTSU Power Supply State Control**

| CTSUCR1.CTSUPON Bit | CTSUSNZ Bit | CTSUCAP Bit | CTSUSTRT Bit | CTSU Power Supply State |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Stopped |
| 1 | 0 | — | — | Operating |
| 1 | 1 | 0 | 0 | Suspended |

Note: Settings other than the above are prohibited.

To start measurement from the suspended state, set the CTSUSNZ bit to 0 and wait for 16 μs before setting the CTSUSTRT bit to 1. To set the suspended state after measurement is finished, set the CTSUSNZ bit to 1.

### CTSUIOC Bit (CTSU Transmit Pin Control)

This bit selects the logic level of the TS pin when the CTSUERRS.CTSUTSOD bit is set to 1.
This bit setting is ignored when the CTSUTSOD bit is 0.

### CTSUINIT Bit (CTSU Control Block Initialization)

The internal control registers can be initialized by writing 1 to this bit. To forcibly stop the current operation, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 simultaneously. In this case, the operation is stopped and the internal control registers are initialized.
Do not write 1 to the CTSUINIT bit at the same time as setting the CTSUSTRT bit to 1 (CTSU operation starts).

## 43.2.2 CTSU Control Register 1 (CTSUCR1)

Address(es): CTSU.CTSUCR1 000A 0901h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CTSUMD[1:0] | | CTSUCLK[1:0] | | CTSUATUNE1 | CTSUATUNE0 | CTSUCSW | CTSUPON |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CTSUPON | CTSU Power Supply Enable | 0: Powered off<br>1: Powered on | R/W |
| b1 | CTSUCSW | CTSU LPF Capacitance Charging Control | 0: Capacitance switch turned off<br>1: Capacitance switch turned on | R/W |
| b2 | CTSUATUNE0 | CTSU Power Supply Operating Mode Setting | VCC ≥ 2.4 V<br>0: Normal operating mode<br>1: Low-voltage operating mode<br>VCC < 2.4 V<br>0: Setting prohibited<br>1: Low-voltage operating mode | R/W |
| b3 | CTSUATUNE1 | CTSU Power Supply Capacity Adjustment | 0: Normal output<br>1: High-current output | R/W |
| b5, b4 | CTSUCLK[1:0] | CTSU Operating Clock Select | b5 b4<br>0 0: PCLK<br>0 1: PCLK/2 (PCLK divided by 2)<br>1 0: PCLK/4 (PCLK divided by 4)<br>1 1: Setting prohibited | R/W |
| b7, b6 | CTSUMD[1:0] | CTSU Measurement Mode Select | b7 b6<br>0 0: Self-capacitance single scan mode<br>0 1: Self-capacitance multi-scan mode<br>1 0: Setting prohibited<br>1 1: Mutual capacitance full scan mode | R/W |

The CTSUCR1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUPON Bit (CTSU Power Supply Enable)**

This bit controls power supply to the CTSU. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

**CTSUCSW Bit (CTSU LPF Capacitance Charging Control)**

This bit controls charging of the LPF capacitor connected to the TSCAP pin (turning on/off of the capacitance switch). After the capacitance switch is turned on, wait until the capacitance connected to the TSCAP pin is charged for the specified time before starting measurement (CTSUCR0.CTSUSTRT = 1). Prior to measurement, use an I/O port to output a low level to the TSCAP pin, and discharge the LPF capacitance that has been already charged. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

**CTSUATUNE0 Bit (CTSU Power Supply Operating Mode Setting)**

This bit sets the power supply operating mode. Set this bit according to the lower limit of the VCC for operating the CTSU. As an example, when performing touch measurement in a system (the VCC voltage range is 2 to 3 V) where the VCC varies depending on battery operation, set this bit to 1 regardless of the initial VCC voltage.

**CTSUATUNE1 Bit (CTSU Power Supply Capacity Adjustment)**

This bit sets the capacity of the CTSU power supply. Normally, the value of this bit should be set to 0.

**CTSUCLK[1:0] Bits (CTSU Operating Clock Select)**

These bits select the operating clock.

**CTSUMD[1:0] Bits (CTSU Measurement Mode Select)**

These bits set the measurement mode. For details, refer to section 43.3.2, Measurement Modes.

## 43.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS)

Address(es): CTSU.CTSUSDPRS 000A 0902h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | CTSUSOFF | CTSUPRMODE[1:0] | | CTSUPRRATIO[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | CTSUPRRATIO[3:0] | CTSU Measurement Time and Pulse Count Adjustment | Recommended setting value: 3 (0011b) | R/W |
| b5, b4 | CTSUPRMODE[1:0] | CTSU Base Period and Pulse Count Setting | b5 b4<br>0 0: 510 pulses<br>0 1: 126 pulses<br>1 0: 62 pulses (recommended setting value)<br>1 1: Setting prohibited | R/W |
| b6 | CTSUSOFF | CTSU High-Pass Noise Reduction Function Off Setting | 0: High-pass noise reduction function turned on<br>1: High-pass noise reduction function turned off | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

The CTSUSDPRS register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUPRRATIO[3:0] Bits (CTSU Measurement Time and Pulse Count Adjustment)**

These bits are used to determine the measurement time and the number of measurement pulses. These are calculated by the following formula. The number of base pulses is determined by setting the CTSUPRMODE[1:0] bits.

Number of measurement pulses = number of base pulses × (CTSUPRRATIO[3:0] bits + 1)

Measurement time = (number of base pulses × (CTSUPRRATIO[3:0] bits + 1) + (number of base pulses – 2) × 0.25) × base clock cycle

Note: For details on the base clock cycle, refer to section 43.2.21, CTSU Sensor Offset Register 1 (CTSUSO1).

**CTSUPRMODE[1:0] Bits (CTSU Base Period and Pulse Count Setting)**

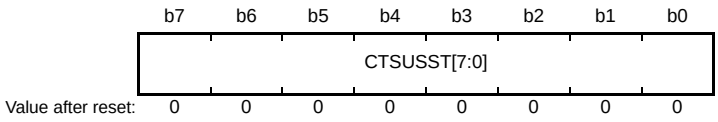
These bits select the number of base pulses during measurement.

**CTSUSOFF Bit (CTSU High-Pass Noise Reduction Function Off Setting)**

This bit turns on or off the function for reducing high-pass noise. Set this bit to 1 when turning off the high-pass noise reduction function.

## 43.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST)

Address(es): CTSU.CTSUSST 000A 0903h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | CTSUSST[7:0] | CTSU Sensor Stabilization Wait Control | The value of these bits should be fixed to 00010000b. | R/W |

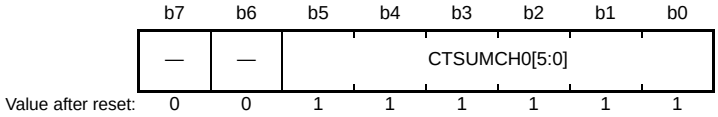
The CTSUSST register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUSST[7:0] Bits (CTSU Sensor Stabilization Wait Control)**

These bits set the stabilization wait time for the TSCAP pin voltage. The value of these bits should be fixed to 00010000b. If these bits are not set, the TSCAP voltage becomes unstable at the start of measurement, and the CTSU is unable to obtain correct touch measurement results.

## 43.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0)

Address(es): CTSU.CTSUMCH0 000A 0904h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | CTSUMCH0[5:0] | CTSU Measurement Channel 0 | • In self-capacitance single scan<br>b5 b0<br>0 0 0 0 1 0: TS2<br>0 0 0 0 1 1: TS3<br>0 0 0 1 0 0: TS4<br>0 0 0 1 1 1: TS7<br>0 0 1 0 0 0: TS8<br>0 0 1 1 0 0: TS12<br>0 0 1 1 0 1: TS13<br>0 1 0 1 1 0: TS22<br>0 1 0 1 1 1: TS23<br>0 1 1 0 1 1: TS27<br>0 1 1 1 1 0: TS30<br>1 0 0 0 1 1: TS35<br>Other than above: Starting measurement operation (CTSUCR0.CTSUSTRT bit = 1) is prohibited after these bits are set.<br><br>• In other measurement modes<br>b5 b0<br>0 0 0 0 1 0: TS2<br>0 0 0 0 1 1: TS3<br>0 0 0 1 0 0: TS4<br>0 0 0 1 1 1: TS7<br>0 0 1 0 0 0: TS8<br>0 0 1 1 0 0: TS12<br>0 0 1 1 0 1: TS13<br>0 1 0 1 1 0: TS22<br>0 1 0 1 1 1: TS23<br>0 1 1 0 1 1: TS27<br>0 1 1 1 1 0: TS30<br>1 0 0 0 1 1: TS35<br>1 1 1 1 1 1: Measurement is stopped | R/W*1 |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Writing to these bits is enabled only in self-capacitance single scan mode (CTSUCR1.CTSUMD[1:0] bits = 00b).

The CTSUMCH0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

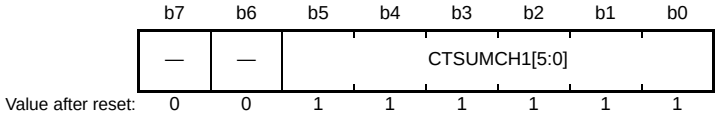
### CTSUMCH0[5:0] Bits (CTSU Measurement Channel 0)

These bits set the channel to be measured in self-capacitance single scan mode, and indicate the receive channel that is being measured in other modes.

Set only enabled channels (000010b, 000011b, 000100b, 000111b, 001000b, 001100b, 001101b, 010110b, 010111b, 011011b, 011110b, 100011b) when setting channels in self-capacitance single scan mode. In other modes, writing to these bits has no effect.

## 43.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1)

Address(es): CTSU.CTSUMCH1 000A 0905h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | CTSUMCH1[5:0] | CTSU Measurement Channel 1 | b5 b0<br>0 0 0 0 1 0: TS2<br>0 0 0 0 1 1: TS3<br>0 0 0 1 0 0: TS4<br>0 0 0 1 1 1: TS7<br>0 0 1 0 0 0: TS8<br>0 0 1 1 0 0: TS12<br>0 0 1 1 0 1: TS13<br>0 1 0 1 1 0: TS22<br>0 1 0 1 1 1: TS23<br>0 1 1 0 1 1: TS27<br>0 1 1 1 1 0: TS30<br>1 0 0 0 1 1: TS35<br>1 1 1 1 1 1: Measurement is stopped | R |
| b7, b6 | — | Reserved | These bits are read as 0. Writing to these bits has no effect. | R |

### CTSUMCH1[5:0] Bits (CTSU Measurement Channel 1)

These bits indicate the transmit channel that is being measured in full scan mode. The value of these bits is 111111b while measurement is stopped or in self-capacitance single scan mode and multi-scan mode.

## 43.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0)

Address(es): CTSU.CTSUCHAC0 000A 0906h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CTSUCHAC07 | — | — | CTSUCHAC04 | CTSUCHAC03 | CTSUCHAC02 | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b2 | CTSUCHAC02 | CTSU Channel 2 Enable Control | 0: Not measurement target<br>1: Measurement target | R/W |
| b3 | CTSUCHAC03 | CTSU Channel 3 Enable Control | | R/W |
| b4 | CTSUCHAC04 | CTSU Channel 4 Enable Control | | R/W |
| b6, b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | CTSUCHAC07 | CTSU Channel 7 Enable Control | 0: Not measurement target<br>1: Measurement target | R/W |

The CTSUCHAC0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHAC0j Bit (CTSU Channel m Enable Control) (j = 2 to 4, 7; m = 2 to 4, 7)**

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

## 43.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1)

Address(es): CTSU.CTSUCHAC1 000A 0907h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | CTSUCHAC15 | CTSUCHAC14 | — | — | — | CTSUCHAC10 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CTSUCHAC10 | CTSU Channel 8 Enable Control | 0: Not measurement target<br>1: Measurement target | R/W |
| b3 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | CTSUCHAC14 | CTSU Channel 12 Enable Control | 0: Not measurement target<br>1: Measurement target | R/W |
| b5 | CTSUCHAC15 | CTSU Channel 13 Enable Control | | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The CTSUCHAC1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHAC1j Bit (CTSU Channel m Enable Control) (j = 0, 4, 5; m = 8, 12, 13)**

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

## 43.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2)

Address(es): CTSU.CTSUCHAC2 000A 0908h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CTSUCHAC27 | CTSUCHAC26 | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | CTSUCHAC26 | CTSU Channel 22 Enable Control | 0: Not measurement target<br>1: Measurement target | R/W |
| b7 | CTSUCHAC27 | CTSU Channel 23 Enable Control | | R/W |

The CTSUCHAC2 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHAC2j Bit (CTSU Channel m Enable Control) (j = 6, 7; m = 22, 23)**

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

## 43.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3)

Address(es): CTSU.CTSUCHAC3 000A 0909h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | CTSUCHAC36 | — | — | CTSUCHAC33 | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3 | CTSUCHAC33 | CTSU Channel 27 Enable Control | 0: Not measurement target<br>1: Measurement target | R/W |
| b5, b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | CTSUCHAC36 | CTSU Channel 30 Enable Control | 0: Not measurement target<br>1: Measurement target | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

The CTSUCHAC3 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHAC3j Bit (CTSU Channel m Enable Control) (j = 3, 6; m = 27, 30)**

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

## 43.2.11 CTSU Channel Enable Control Register 4 (CTSUCHAC4)

Address(es): CTSU.CTSUCHAC4 000A 090Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CTSUCHAC43 | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3 | CTSUCHAC43 | CTSU Channel 35 Enable Control | 0: Not measurement target<br>1: Measurement target | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The CTSUCHAC4 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHAC43 Bit (CTSU Channel 35 Enable Control)**

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

## 43.2.12 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0)

Address(es): CTSU.CTSUCHTRC0 000A 090Bh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CTSUCHTRC07 | — | — | CTSUCHTRC04 | CTSUCHTRC03 | CTSUCHTRC02 | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b2 | CTSUCHTRC02 | CTSU Channel 2 Transmit/Receive Control | 0: Reception<br>1: Transmission | R/W |
| b3 | CTSUCHTRC03 | CTSU Channel 3 Transmit/Receive Control | | R/W |
| b4 | CTSUCHTRC04 | CTSU Channel 4 Transmit/Receive Control | | R/W |
| b6, b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | CTSUCHTRC07 | CTSU Channel 7 Transmit/Receive Control | 0: Reception<br>1: Transmission | R/W |

The CTSUCHTRC0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHTRC0j Bit (CTSU Channel m Transmit/Receive Control) (j = 2 to 4, 7; m = 2 to 4, 7)**

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

## 43.2.13 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1)

Address(es): CTSU.CTSUCHTRC1 000A 090Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | CTSUCHTRC15 | CTSUCHTRC14 | — | — | — | CTSUCHTRC10 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CTSUCHTRC10 | CTSU Channel 8 Transmit/Receive Control | 0: Reception<br>1: Transmission | R/W |
| b3 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | CTSUCHTRC14 | CTSU Channel 12 Transmit/Receive Control | 0: Reception<br>1: Transmission | R/W |
| b5 | CTSUCHTRC15 | CTSU Channel 13 Transmit/Receive Control | | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The CTSUCHTRC1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHTRC1j Bit (CTSU Channel m Transmit/Receive Control) (j = 0, 4, 5; m = 8, 12, 13)**

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

## 43.2.14 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2)

Address(es): CTSU.CTSUCHTRC2 000A 090Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CTSUCHTRC27 | CTSUCHTRC26 | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | CTSUCHTRC26 | CTSU Channel 22 Transmit/Receive Control | 0: Reception<br>1: Transmission | R/W |
| b7 | CTSUCHTRC27 | CTSU Channel 23 Transmit/Receive Control | | R/W |

The CTSUCHTRC2 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHTRC2j Bit (CTSU Channel m Transmit/Receive Control) (j = 6, 7; m = 22, 23)**

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

## 43.2.15 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3)

Address(es): CTSU.CTSUCHTRC3 000A 090Eh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | CTSUCHTRC36 | — | — | CTSUCHTRC33 | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3 | CTSUCHTRC33 | CTSU Channel 27 Transmit/Receive Control | 0: Reception<br>1: Transmission | R/W |
| b5, b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | CTSUCHTRC36 | CTSU Channel 30 Transmit/Receive Control | 0: Reception<br>1: Transmission | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

The CTSUCHTRC3 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHTRC3j Bit (CTSU Channel m Transmit/Receive Control) (j = 3, 6; m = 27, 30)**

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

## 43.2.16 CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4)

Address(es): CTSU.CTSUCHTRC4 000A 090Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CTSUCHTRC43 | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3 | CTSUCHTRC43 | CTSU Channel 35 Transmit/Receive Control | 0: Reception<br>1: Transmission | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The CTSUCHTRC4 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUCHTRC43 Bit (CTSU Channel 35 Transmit/Receive Control)**

This bit allocates reception or transmission to the corresponding TS35 pin in full scan mode. The setting of this bit is ignored in self-capacitance single scan mode and multi-scan mode.

## 43.2.17 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC)

Address(es): CTSU.CTSUDCLKC 000A 0910h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | CTSUSSCNT[1:0] | | — | — | CTSUSSMOD[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CTSUSSMOD[1:0] | CTSU Diffusion Clock Mode Select | These bits should be set to 00b. | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b5, b4 | CTSUSSCNT[1:0] | CTSU Diffusion Clock Control | These bits should be set to 11b. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The CTSUDCLKC register should be set when the CTSUCR0.CTSUSTRT bit is 0.

**CTSUSSMOD[1:0] Bits (CTSU Diffusion Clock Mode Select)**

These bits set the mode of the spectrum diffusion clock for high-pass noise reduction. When using the high-pass function, the value of these bits should be fixed to 00b. If these bits are not set, the effect of high-pass noise reduction cannot be correctly obtained.

**CTSUSSCNT[1:0] Bits (CTSU Diffusion Clock Control)**

These bits adjust the spectrum diffusion amount to reduce high-pass noise. When using the high-pass noise reduction function, the value of these bits should be fixed to 11b. If these bits are not set, touch measurement may not be correctly performed.

## 43.2.18 CTSU Status Register (CTSUST)

Address(es): CTSU.CTSUST 000A 0911h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CTSUPS | CTSUROVF | CTSUSOVF | CTSUDTSR | — | CTSUSTC[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | CTSUSTC[2:0] | CTSU Measurement Status Counter | b2 b1 b0<br>0 0 0: Status 0<br>0 0 1: Status 1<br>0 1 0: Status 2<br>0 1 1: Status 3<br>1 0 0: Status 4<br>1 0 1: Status 5 | R |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b4 | CTSUDTSR | CTSU Data Transfer Status Flag | 0: Measurement result has been read<br>1: Measurement result has not been read | R |
| b5 | CTSUSOVF | CTSU Sensor Counter Overflow Flag | 0: No overflow<br>1: An overflow | R/W |
| b6 | CTSUROVF | CTSU Reference Counter Overflow Flag | 0: No overflow<br>1: An overflow | R/W |
| b7 | CTSUPS | CTSU Mutual Capacitance Status Flag | 0: First measurement<br>1: Second measurement | R |

When using the CTSUCR0.CTSUINIT bit to clear an overflow flag, make sure that the CTSUCR0.CTSUSTRT bit is 0.

### CTSUSTC[2:0] Flags (CTSU Measurement Status Counter)

These counters indicate the current measurement status. For details on each status, refer to section 43.3.2.2, Status Counter.

### CTSUDTSR Flag (CTSU Data Transfer Status Flag)

This flag indicates whether the measurement result stored in the sensor counter and the reference counter has been read. This flag is set to 1 when measurement is completed; 0 when the reference counter is read by software or the DTC. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

### CTSUSOVF Flag (CTSU Sensor Counter Overflow Flag)

This flag indicates whether the sensor counter has overflowed. FFFFh can be read as the measurement result (CTSUSC counter) when an overflow has occurred.

Even if an overflow occurs, measurement processing is continued until the set period.

No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).

This flag is cleared when 0 is written after 1 is read by software. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

### CTSUROVF Flag (CTSU Reference Counter Overflow Flag)

This flag indicates whether the reference counter has overflowed. FFFFh can be read as the measurement result (CTSURC counter) when an overflow has occurred.

Even if an overflow occurs, measurement processing is continued until the set period.

No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).
This flag is cleared when 0 is written after 1 is read by software. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

**CTSUPS Flag (CTSU Mutual Capacitance Status Flag)**

This flag indicates whether the measurement is the first or second of two measurements for each channel in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).
This flag indicates 0 while measurement is stopped or in other measurement modes.

## 43.2.19 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC)

Address(es): CTSU.CTSUSSC 000A 0912h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CTSUSSDIV[3:0] | | | | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 to b8 | CTSUSSDIV[3:0] | CTSU Spectrum Diffusion Frequency Division Setting | These bits specify the spectrum diffusion frequency division setting according to the base clock frequency division setting. | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### CTSUSSDIV[3:0] Bits (CTSU Spectrum Diffusion Frequency Division Setting)

These bits specify the spectrum diffusion frequency division setting according to the base clock frequency division setting. See the relationship between base clock frequencies and the CTSUSSDIV[3:0] bits settings in Table 43.5, for setting the value of these bits.

**Table 43.5 Relationship between Base Clock Frequencies and the CTSUSSDIV[3:0] Bits Settings**

| Base Clock Frequency fb (MHz) | CTSUSSDIV[3:0] Bits Setting |
|---|---|
| 4.00 ≤ fb | 0000b |
| 2.00 ≤ fb < 4.00 | 0001b |
| 1.33 ≤ fb < 2.00 | 0010b |
| 1.00 ≤ fb < 1.33 | 0011b |
| 0.80 ≤ fb < 1.00 | 0100b |
| 0.67 ≤ fb < 0.80 | 0101b |
| 0.57 ≤ fb < 0.67 | 0110b |
| 0.50 ≤ fb < 0.57 | 0111b |
| 0.44 ≤ fb < 0.50 | 1000b |
| 0.40 ≤ fb < 0.44 | 1001b |
| 0.36 ≤ fb < 0.40 | 1010b |
| 0.33 ≤ fb < 0.36 | 1011b |
| 0.31 ≤ fb < 0.33 | 1100b |
| 0.29 ≤ fb < 0.31 | 1101b |
| 0.27 ≤ fb < 0.29 | 1110b |
| fb < 0.27 | 1111b |

## 43.2.20 CTSU Sensor Offset Register 0 (CTSUSO0)

Address(es): CTSU.CTSUSO0 000A 0914h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CTSUSNUM[5:0] | | | | | | CTSUSO[9:0] | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b9 to b0 | CTSUSO[9:0] | CTSU Sensor Offset Adjustment | b9 b0<br>0 0 0 0 0 0 0 0 0 0: Current offset amount is 0<br>0 0 0 0 0 0 0 0 0 1: Current offset amount is 1<br>0 0 0 0 0 0 0 0 1 0: Current offset amount is 2<br>: :<br>1 1 1 1 1 1 1 1 1 0: Current offset amount is 1022<br>1 1 1 1 1 1 1 1 1 1: Current offset amount is maximum | R/W |
| b15 to b10 | CTSUSNUM[5:0] | CTSU Measurement Count Setting | These bits set the number of measurements. | R/W |

### CTSUSO[9:0] Bits (CTSU Sensor Offset Adjustment)

These control bits adjust the input current offset of the sensor ICO. These bits are used to offset the sensor ICO input current generated from electrostatic capacitance while the electrode is not being touched during touch measurement, thus preventing overflow of the CTSU sensor counter.
Make settings for the TS pin that is to be measured next after a CTSUWR interrupt is generated.

### CTSUSNUM[5:0] Bits (CTSU Measurement Count Setting)

These bits set how many times the number of measurement pulses specified by the CTSUSDPRS.CTSUPRRATIO[3:0] and CTSUSDPRS.CTSUPRMODE[1:0] bits is repeated in the measurement time. The number of measurement pulses is repeated (CTSUSNUM[5:0] bits + 1) times.
Make settings for the TS pin that is to be measured next after a CTSUWR interrupt is generated.

## 43.2.21 CTSU Sensor Offset Register 1 (CTSUSO1)

Address(es): CTSU.CTSUSO1 000A 0916h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | CTSUICOG[1:0] | | CTSUSDPA[4:0] | | | | | CTSURICOA[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | CTSURICOA[7:0] | CTSU Reference ICO Current Adjustment | b7 b0<br>0 0 0 0 0 0 0 0: Input current amount 0<br>0 0 0 0 0 0 0 1: Input current amount 1<br>0 0 0 0 0 0 1 0: Input current amount 2<br>: :<br>1 1 1 1 1 1 1 0: Input current amount 254<br>1 1 1 1 1 1 1 1: Input current amount maximum | R/W |
| b12 to b8 | CTSUSDPA[4:0] | CTSU Base Clock Setting | b12 b8<br>0 0 0 0 0: Operating clock divided by 2*1<br>0 0 0 0 1: Operating clock divided by 4<br>0 0 0 1 0: Operating clock divided by 6<br>0 0 0 1 1: Operating clock divided by 8<br>0 0 1 0 0: Operating clock divided by 10<br>0 0 1 0 1: Operating clock divided by 12<br>0 0 1 1 0: Operating clock divided by 14<br>0 0 1 1 1: Operating clock divided by 16<br>0 1 0 0 0: Operating clock divided by 18<br>0 1 0 0 1: Operating clock divided by 20<br>0 1 0 1 0: Operating clock divided by 22<br>0 1 0 1 1: Operating clock divided by 24<br>0 1 1 0 0: Operating clock divided by 26<br>0 1 1 0 1: Operating clock divided by 28<br>0 1 1 1 0: Operating clock divided by 30<br>0 1 1 1 1: Operating clock divided by 32<br>1 0 0 0 0: Operating clock divided by 34<br>1 0 0 0 1: Operating clock divided by 36<br>1 0 0 1 0: Operating clock divided by 38<br>1 0 0 1 1: Operating clock divided by 40<br>1 0 1 0 0: Operating clock divided by 42<br>1 0 1 0 1: Operating clock divided by 44<br>1 0 1 1 0: Operating clock divided by 46<br>1 0 1 1 1: Operating clock divided by 48<br>1 1 0 0 0: Operating clock divided by 50<br>1 1 0 0 1: Operating clock divided by 52<br>1 1 0 1 0: Operating clock divided by 54<br>1 1 0 1 1: Operating clock divided by 56<br>1 1 1 0 0: Operating clock divided by 58<br>1 1 1 0 1: Operating clock divided by 60<br>1 1 1 1 0: Operating clock divided by 62<br>1 1 1 1 1: Operating clock divided by 64 | R/W |
| b14, b13 | CTSUICOG[1:0] | CTSU ICO Gain Adjustment | b14 b13<br>0 0: 100% gain<br>0 1: 66% gain<br>1 0: 50% gain<br>1 1: 40% gain | R/W |
| b15 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note 1. The CTSUSDPA[4:0] bits should not be set to 00000b while the high-pass noise reduction function is turned off (CTSUSDPRS.CTSUSOFF bit = 1) in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).

Write first to the CTSUSSC register, then CTSUSO0 register, and then CTSUSO1 register after a CTSUWR interrupt is generated. Write operation to the CTSUSO1 register causes a transition to Status 3. Thus, set all the bits in a single setting when writing to the CTSUSO1 register.

**CTSURICOA[7:0] Bits (CTSU Reference ICO Current Adjustment)**

These bits adjust the oscillation frequency using the input current of the reference ICO.

**CTSUSDPA[4:0] Bits (CTSU Base Clock Setting)**

These bits are used to generate a base clock used as the source for the sensor drive pulse by dividing the operating clock. For details on the setting procedure, refer to section 43.3.2.1, Initial Setting Flowchart.

**CTSUICOG[1:0] Bits (CTSU ICO Gain Adjustment)**

These bits adjust the output frequency gain of the sensor ICO and the reference ICO. If changes in the capacitance between when the electrode is touched and when it is not touched greatly exceed the dynamic range of the sensor ICO, set the gain adjustment bits to adjust the gain appropriately.

## 43.2.22 CTSU Sensor Counter (CTSUSC)

Address(es): CTSU.CTSUSC 000A 0918h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CTSUSC[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | CTSUSC[15:0] | CTSU Sensor Counter | These bits indicate FFFFh when an overflow occurs. | R |

Read first from the CTSUSC counter and then the CTSURC counter after a CTSURD interrupt is generated.

**CTSUSC[15:0] Bits (CTSU Sensor Counter)**

These bits are configured as an increment counter that counts the sensor ICO clock.
Read these bits after a CTSURD interrupt is generated. After the CTSURC counter is read, these bits are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags changes to 100b) in the next measurement. These bits are also cleared using the CTSUCR0.CTSUINIT bit.

## 43.2.23 CTSU Reference Counter (CTSURC)

Address(es): CTSU.CTSURC 000A 091Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CTSURC[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | CTSURC[15:0] | CTSU Reference Counter | These bits indicate FFFFh when an overflow occurs. | R |

Read first from the CTSUSC counter and then the CTSURC counter after a CTSURD interrupt is generated.
Even when the stabilization time specified for Status 3 has elapsed, if the CTSURC counter is not read, Status 3 continues until the counter is read.

### CTSURC[15:0] Bits (CTSU Reference Counter)

These bits are configured as an increment counter that counts the reference ICO clock.
The reference ICO is used to optimize touch measurement performed using the sensor ICO. There is some deviation depending on the internal sensor ICO and the reference ICO in the CTSU, but both ICOs have almost the same characteristics, and the dynamic range and the current to frequency characteristics are almost the same. The range of current amount that can be set by the reference ICO current adjustment bits is about the same as the range of both ICOs, and the current amount input to the sensor ICO must be within this dynamic range. First, use the reference ICO to check the differences between the ICOs and measure the current to oscillation frequency characteristics. Since the reference ICO oscillation frequency can be obtained from the reference ICO counter, the ICO oscillation frequency (counter value/ measurement time) for the input current amount can be measured by setting the value in the reference ICO current adjustment bits and measuring the reference ICO counter. The reference ICO counter value measured using the maximum value of the reference ICO current adjustment bits is the maximum value of the ICO dynamic range. Therefore, the current amount of the sensor ICO needs to be offset by setting the offset adjustment bits so that the sensor ICO counter value does not exceed this value.
Read the CTSURC[15:0] bits after a CTSURD interrupt is generated. After these bits are read, they are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags changes to 100b) in the next measurement. These bits are also cleared using the CTSUCR0.CTSUINIT bit.

## 43.2.24 CTSU Error Status Register (CTSUERRS)

Address(es): CTSU.CTSUERRS 000A 091Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CTSUICOMP | — | — | — | — | — | — | — | CTSUTSOC | — | — | — | CTSUDRV | CTSUTSOD | CTSUSPMD[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CTSUSPMD[1:0] | Calibration Mode | b1 b0<br>0 0: Capacitance measurement mode<br>0 1: Setting prohibited<br>1 0: Calibration mode<br>1 1: Setting prohibited | R/W |
| b2 | CTSUTSOD | TS Pin Fixed Output | 0: Capacitance measurement mode<br>1: TS pins are forced to be high or low | R/W |
| b3 | CTSUDRV | Calibration Setting 1 | 0: Capacitance measurement mode<br>1: Calibration setting 1 | R/W |
| b6 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | CTSUTSOC | Calibration Setting 2 | 0: Capacitance measurement mode<br>1: Calibration setting 2 | R/W |
| b14 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 | CTSUICOMP | TSCAP Voltage Error Monitor | 0: Normal TSCAP voltage<br>1: Abnormal TSCAP voltage | R |

### CTSUSPMD[1:0] Bit (Calibration Mode)

These bits are used to calibrate the CTSU.
When measuring the capacitance, set these bits to 00b.

### CTSUTSOD Bit (TS Pin Fixed Output)

This bit is used to calibrate the CTSU. When setting this bit to 1, the TS pins are forced to the logic level specified by the CTSUCR0.CTSUIOC bit.
When measuring the capacitance, set this bit to 0.

### CTSUDRV Bit (Calibration Setting 1)

This bit is used to calibrate the CTSU.
When measuring the capacitance, set this bit to 0.

### CTSUTSOC Bit (Calibration Setting 2)

This bit is used to calibrate the CTSU.
When measuring the capacitance, set this bit to 0.

### CTSUICOMP Bit (TSCAP Voltage Error Monitor)

If the offset current amount set by the CTSUSO1 register exceeds the sensor ICO input current during touch measurement, the TSCAP voltage becomes abnormal and touch measurement cannot be correctly performed. This bit monitors the TSCAP voltage and it is set to 1 if the voltage becomes abnormal. If the TSCAP voltage becomes abnormal, the sensor ICO counter value will be undefined, but touch measurement is normally completed, so it difficult to detect an abnormality by reading the sensor ICO counter value. If the CTSU reference ICO current adjustment bits (CTSURICOA[7:0]) in the CTSUSO1 register are set to a value other than 0, check this bit when touch measurement is completed.

This bit is cleared by writing 0 to the CTSUCR1.CTSUPON bit and turning off the power supply.

## 43.3 Operation

### 43.3.1 Principles of Measurement Operation

Figure 43.4 shows the measurement circuit.

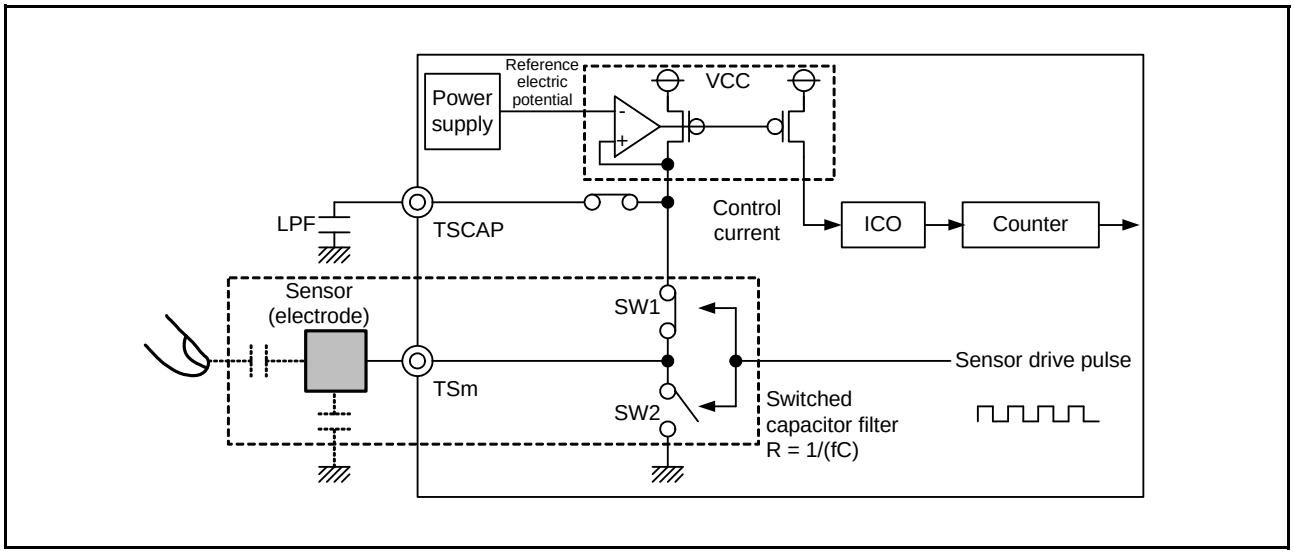



**Figure 43.4 Measurement Circuit (m = 2, 3, 4, 7, 8, 12, 13, 22, 23, 27, 30, 35)**

The electrostatic capacitance measurement operation principles of the CTSU current frequency conversion method are explained using Figure 43.5 to Figure 43.7.

(1) The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (Figure 43.5).

(2) The charged capacitance is discharged by turning SW1 off and SW2 on (Figure 43.6).

Current flows to the switched capacitor filter by switching between charging and discharging in steps (1) and (2). At this time, the value of electrostatic capacitance varies depending on whether a finger is in close proximity, so the flowing current changes. A clock is generated by supplying the control current, which is proportional to the amount of the current flowing through the switched capacitor filter, from the circuit that generates the TSCAP power supply to the ICO. The counter is used to measure the clock frequency which changes depending on whether a finger is in close proximity, and the value read from the counter is used by software to determine contact with a finger (Figure 43.7).

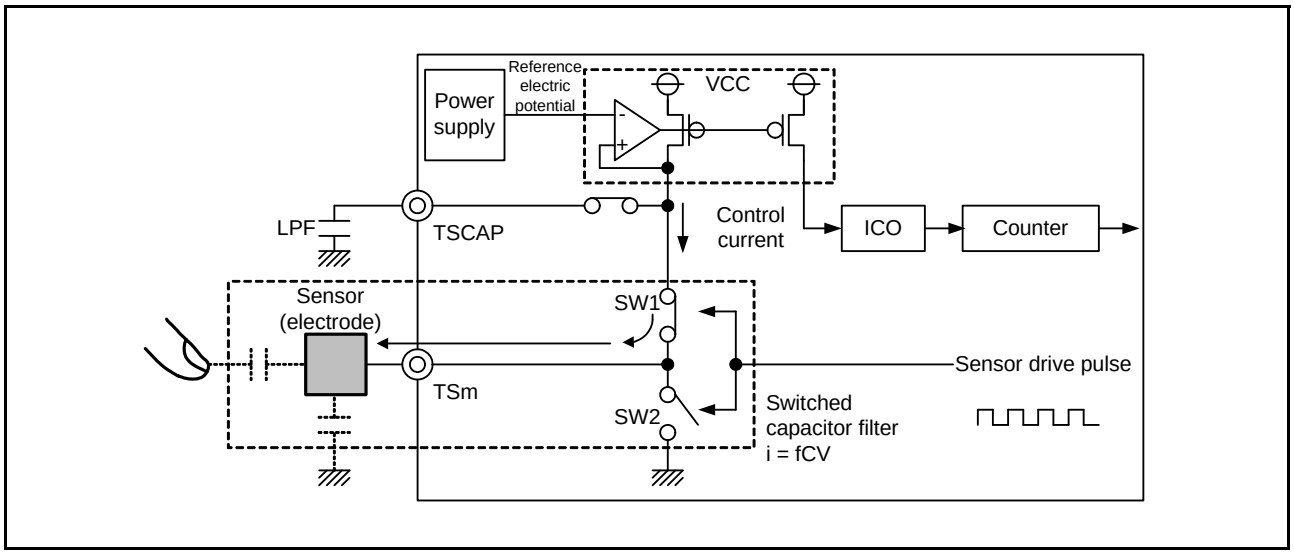



**Figure 43.5 Charging Operation (m = 2, 3, 4, 7, 8, 12, 13, 22, 23, 27, 30, 35)**

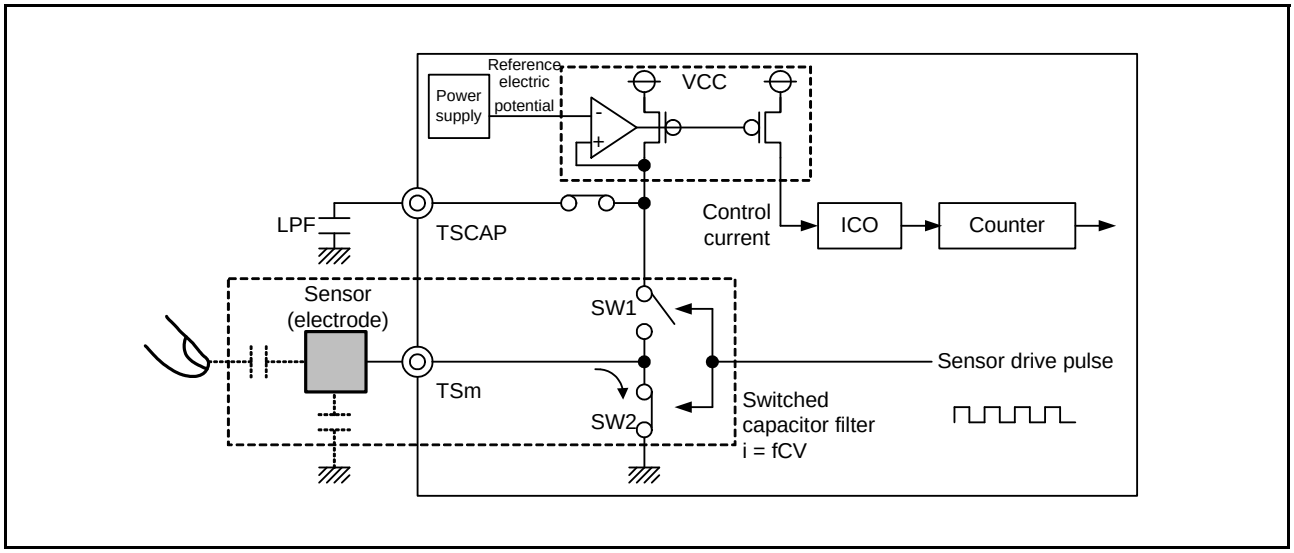



**Figure 43.6 Discharging Operation (m = 2, 3, 4, 7, 8, 12, 13, 22, 23, 27, 30, 35)**

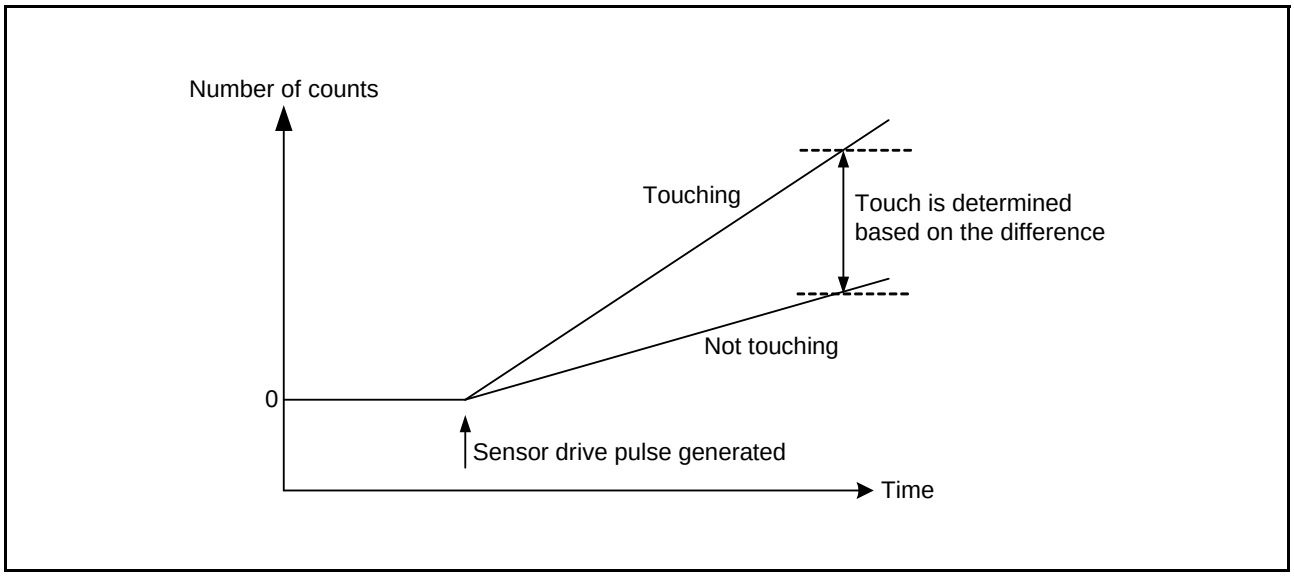



**Figure 43.7 Change in Measured Value When Finger is Touching and Not Touching**

### 43.3.2 Measurement Modes

The CTSU supports self-capacitance and mutual capacitance methods. Figure 43.8 illustrates these methods.

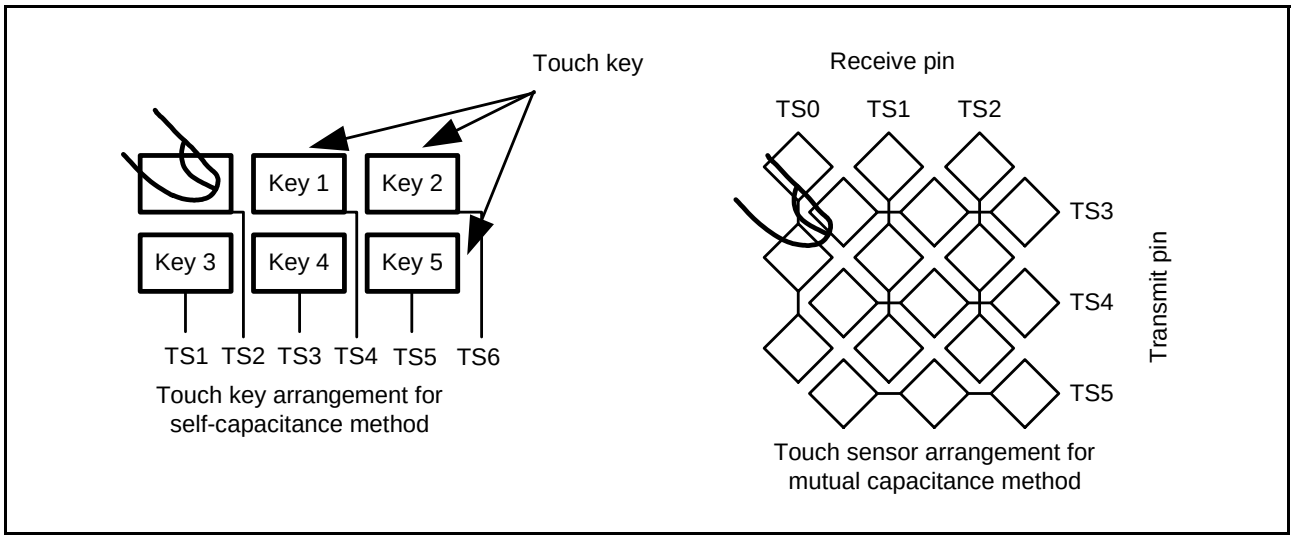



**Figure 43.8 Overview of Self-Capacitance Method and Mutual Capacitance Method**

In the self-capacitance method, a single touch pin is allocated to a single touch key to measure individual electrostatic capacitance when a finger is in close proximity. In this method, single scan and multi-scan can be used as measurement modes.

In the mutual capacitance method, the capacitance between two opposite electrodes (transmit and receive pins) is measured.

### 43.3.2.1 Initial Setting Flowchart

Figure 43.9 shows the flowchart for CTSU initial setting.

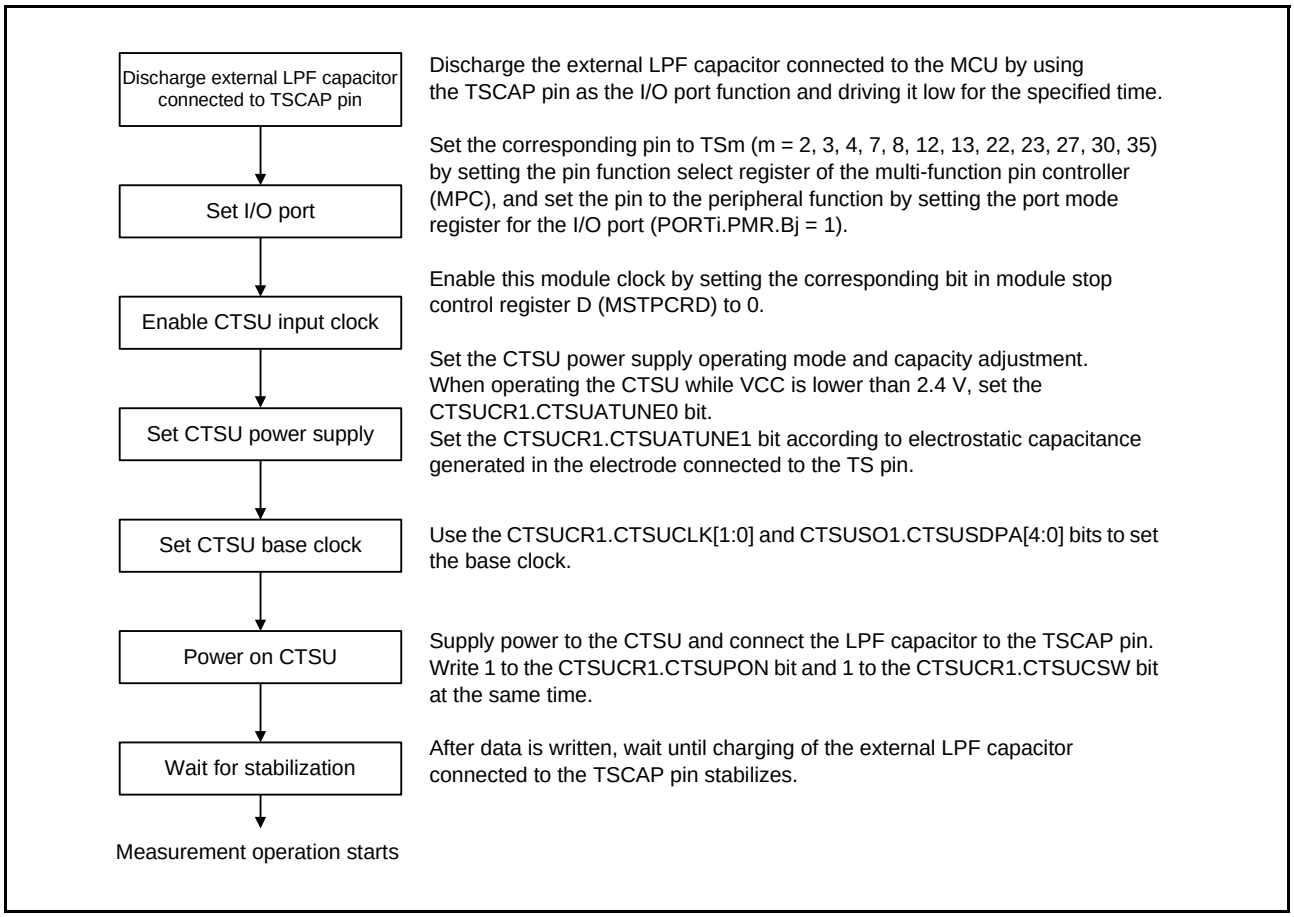



**Figure 43.9 CTSU Initial Setting Flowchart**

Figure 43.10 shows the flowchart for stopping CTSU operation and setting to the standby state.

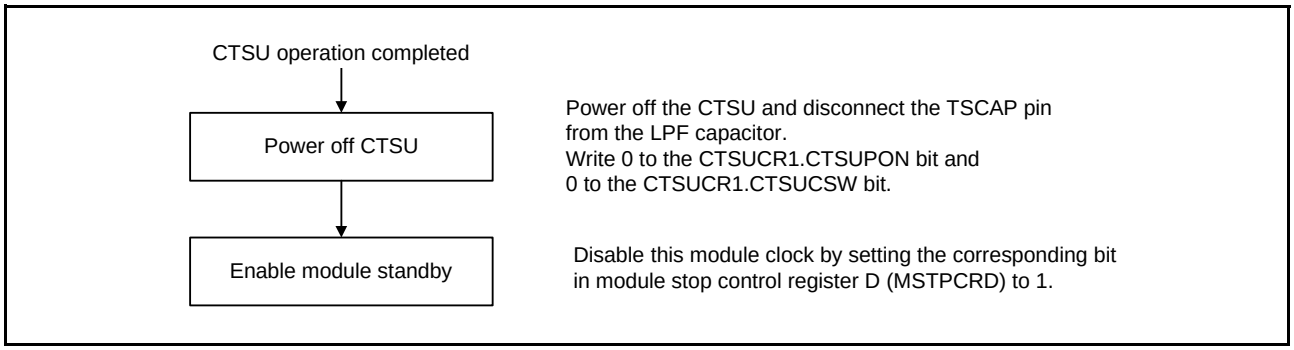



**Figure 43.10 CTSU Stopping Flowchart**

When restarting operation after it has been stopped, follow the initial setting flowchart shown in Figure 43.9.

### 43.3.2.2 Status Counter

The measurement status counter of the CTSU status register (CTSUST) indicates the current measurement status. The measurement status is common to all four modes. Figure 43.11 shows status operation transitions.

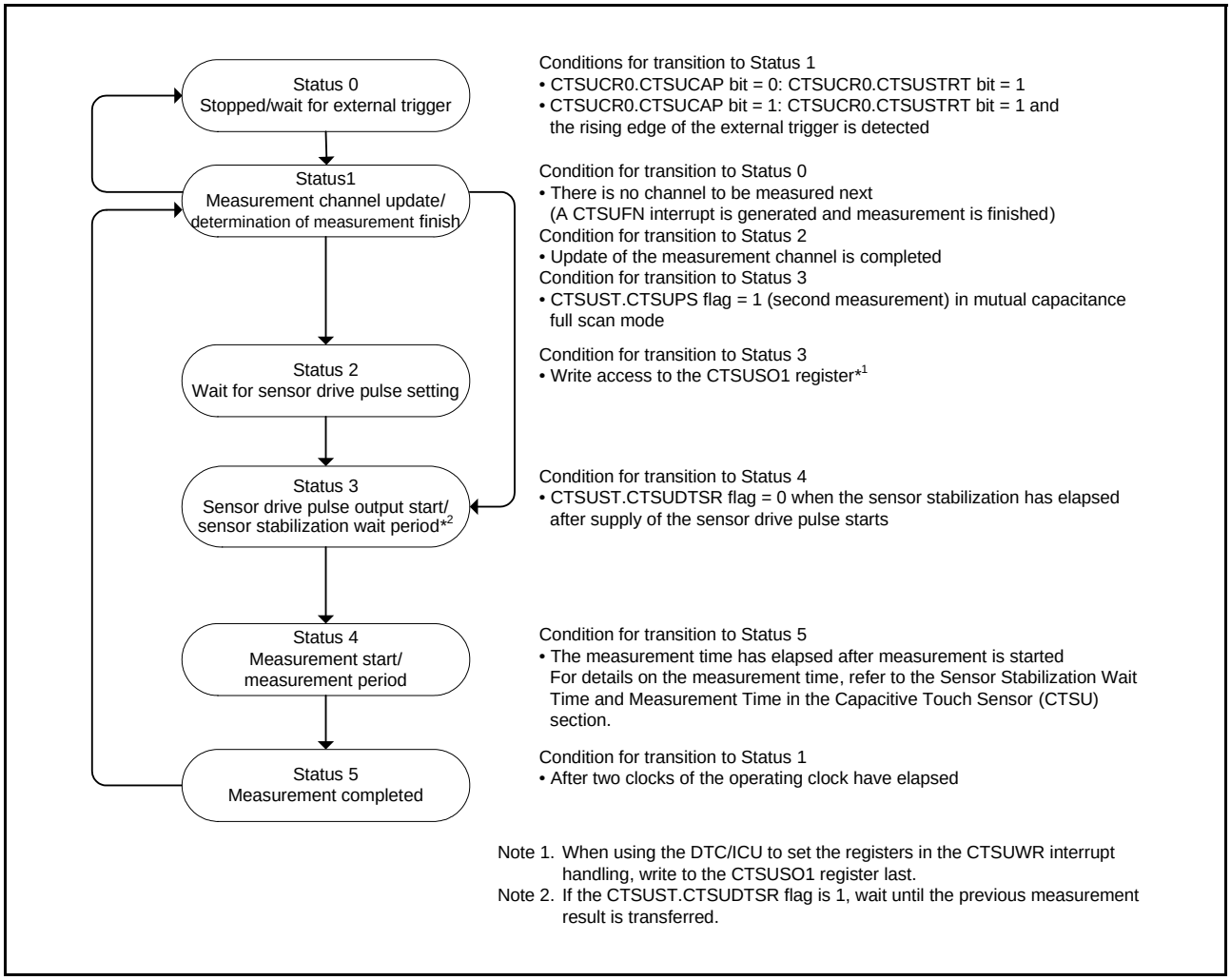



**Figure 43.11 Status Operation Transitions**

The status of the status counter transitions to Status 0 when all of the specified measurement channels are measured. The CTSUCR0.CTSUSTRT bit is cleared to 0 by hardware when a software trigger is used. When an external trigger is used, the value 1 is retained, and the CTSU waits for the next trigger.

When operation is forcibly stopped (by writing 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time) during measurement or the wait state for the trigger, the status transitions to Status 0 and measurement is stopped forcibly.

If there is no channel to be measured by setting the CTSUMCH0, CTSUCHACn, and CTSUCHTRCn registers (n = 0 to 4), a CTSUFN interrupt is generated immediately after a transition to Status 1, and then the status transitions to Status 0. The following are the cases when there is no channel to be measured.

- A measurement target channel is not specified by the CTSUCHACn registers.
- In self-capacitance single scan mode, the channel specified in the CTSUMCH0 register is not a measurement target in the CTSUCHACn registers.
- In full scan mode, there is no transmit channel or receive channel to be measured by combining the CTSUCHACn and CTSUCHTRCn registers.

### 43.3.2.3 Self-Capacitance Single Scan Mode Operation

In self-capacitance single scan mode, electrostatic capacitance on a channel is measured. Figure 43.12 shows the software flowchart and an operation example, and Figure 43.13 shows the timing chart.

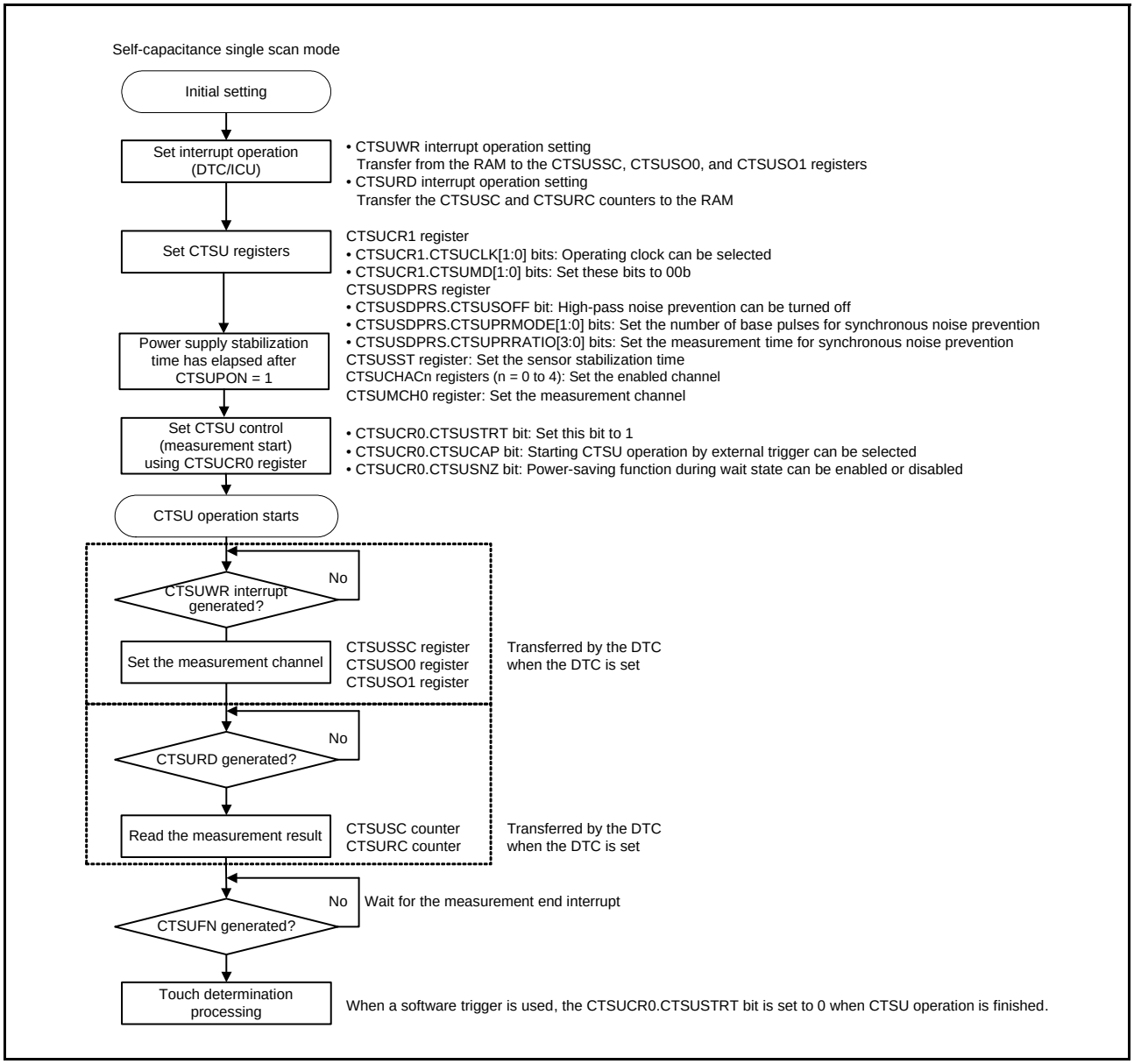



**Figure 43.12 Software Flowchart and Operation Example of Self-Capacitance Single Scan Mode**

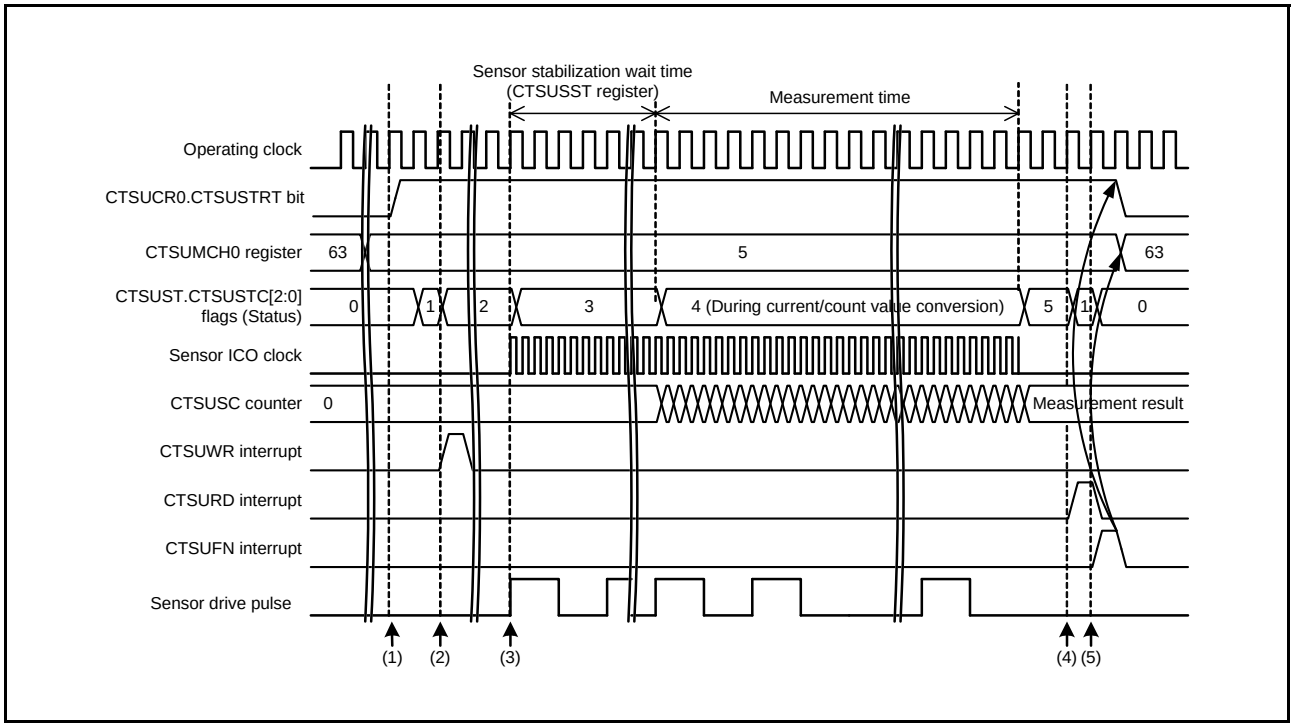



**Figure 43.13 Timing Chart of Self-Capacitance Single Scan Mode (Measurement Start Condition is Software Trigger)**

The following describes operation shown in the timing chart in Figure 43.13.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) A measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

Table 43.6 lists the touch pin states in self-capacitance single scan mode.

**Table 43.6 Touch Pin States in Self-Capacitance Single Scan Mode**

| | Touch Pin | |
|---|---|---|
| **Status** | **Measurement Channel** | **Non-Measurement Channel** |
| 0 | Low | Low |
| 1 | Low | Low |
| 2 | Low | Low |
| 3 | Pulse | Low |
| 4 | Pulse | Low |
| 5 | Low | Low |

### 43.3.2.4 Self-Capacitance Multi-Scan Mode Operation

In self-capacitance multi-scan mode, electrostatic capacitance on all channels that are specified as measurement targets by setting the CTSUCHACn registers (n = 0 to 4) are measured sequentially in ascending order. Figure 43.14 shows the software flowchart and an operation example, and Figure 43.15 shows the timing chart.

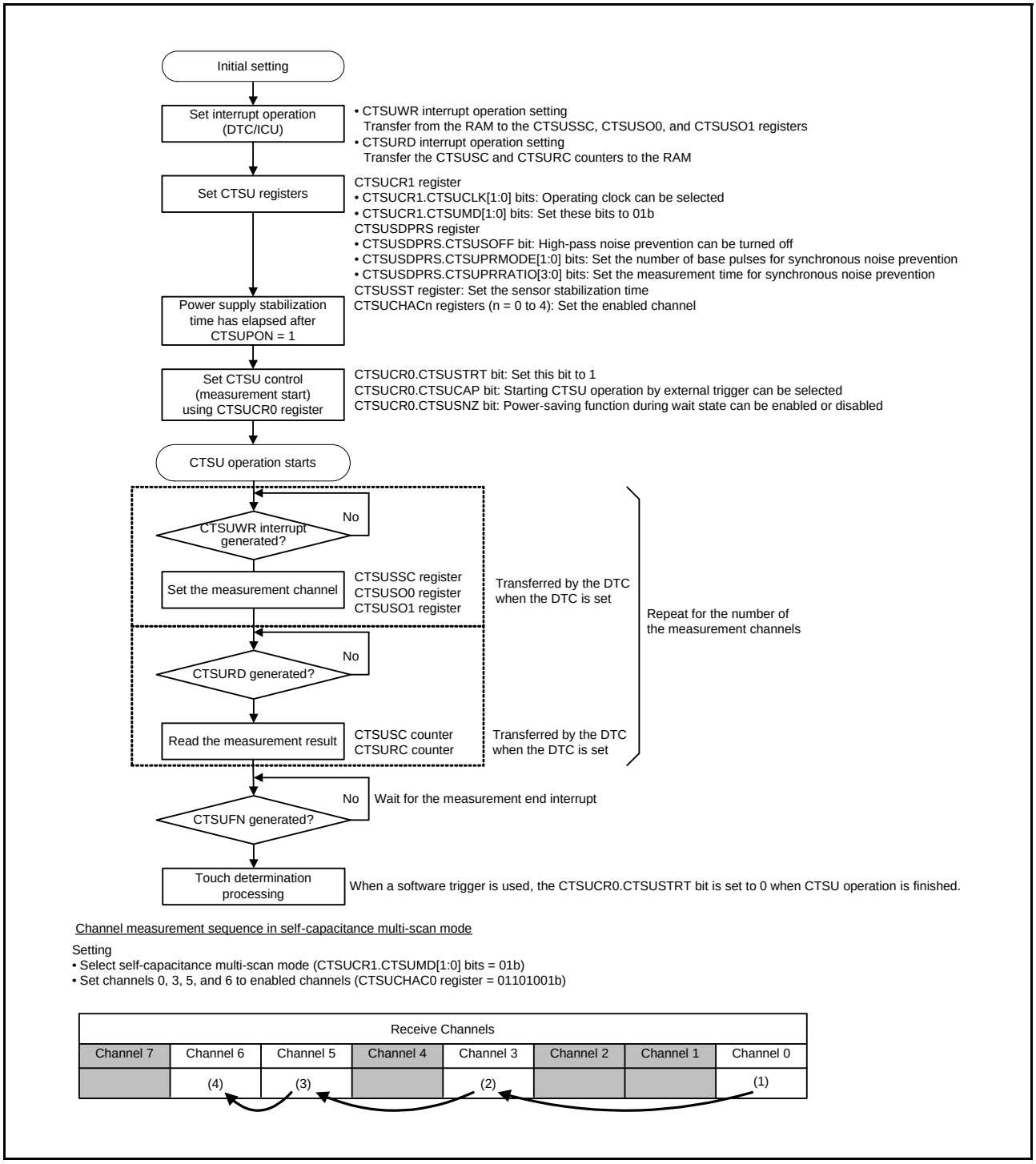



**Figure 43.14 Software Flow and Operation Example of Self-Capacitance Multi-Scan Mode**

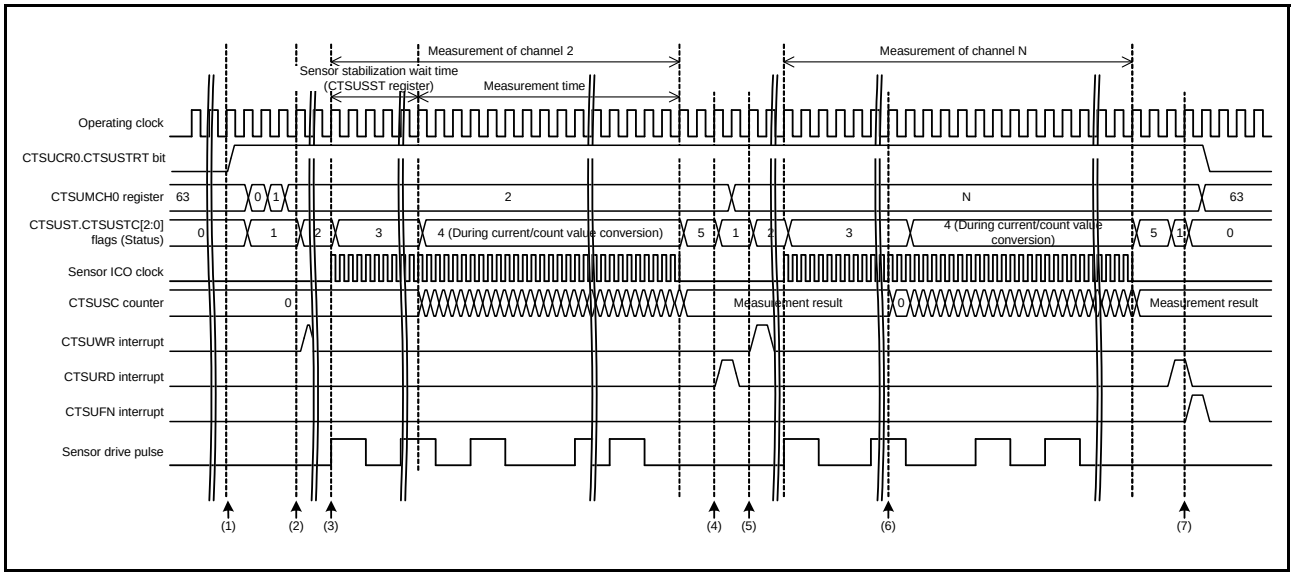



**Figure 43.15 Timing Chart of Self-Capacitance Multi-Scan Mode (Measurement Start Condition is Software Trigger)**

The following describes operation shown in the timing chart in Figure 43.15.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) After a channel to be measured next is determined, a measurement channel setting request (CTSUWR) is output.

(6) After the stabilization wait time has elapsed and when the previous measurement is read, the result is cleared and measurement is started.

(7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

Table 43.7 lists the touch pin states in self-capacitance multi-scan mode.

**Table 43.7 Touch Pin States in Self-Capacitance Multi-Scan Mode**

| Status | Touch Pin | |
|---|---|---|
| | Measurement Channel | Non-Measurement Channel |
| 0 | Low | Low |
| 1 | Low | Low |
| 2 | Low | Low |
| 3 | Pulse | Low |
| 4 | Pulse | Low |
| 5 | Low | Low |

### 43.3.2.5 Mutual Capacitance Full Scan Mode Operation

In mutual capacitance full scan mode, measurement is performed during the high-level period of the sensor drive pulse on the receive channel by applying the edge to the target transmit channel to be measured. A single measurement target is measured twice, at the rising and falling edges. The difference between the data of these two measurements is used to determine whether or not the electrode is touched, thus achieving higher touch sensitivity.
Electrostatic capacitance is measured sequentially on channels set to transmission or reception specified by the CTSUCHTRCn registers (n = 0 to 4), and measurement targets specified by the CTSUCHACn registers. Electrostatic capacitance is measured by combining signals from the measurement target pins that are allocated to transmission or reception. Figure 43.16 shows the software flowchart and an operation example, and Figure 43.17 shows the timing chart.

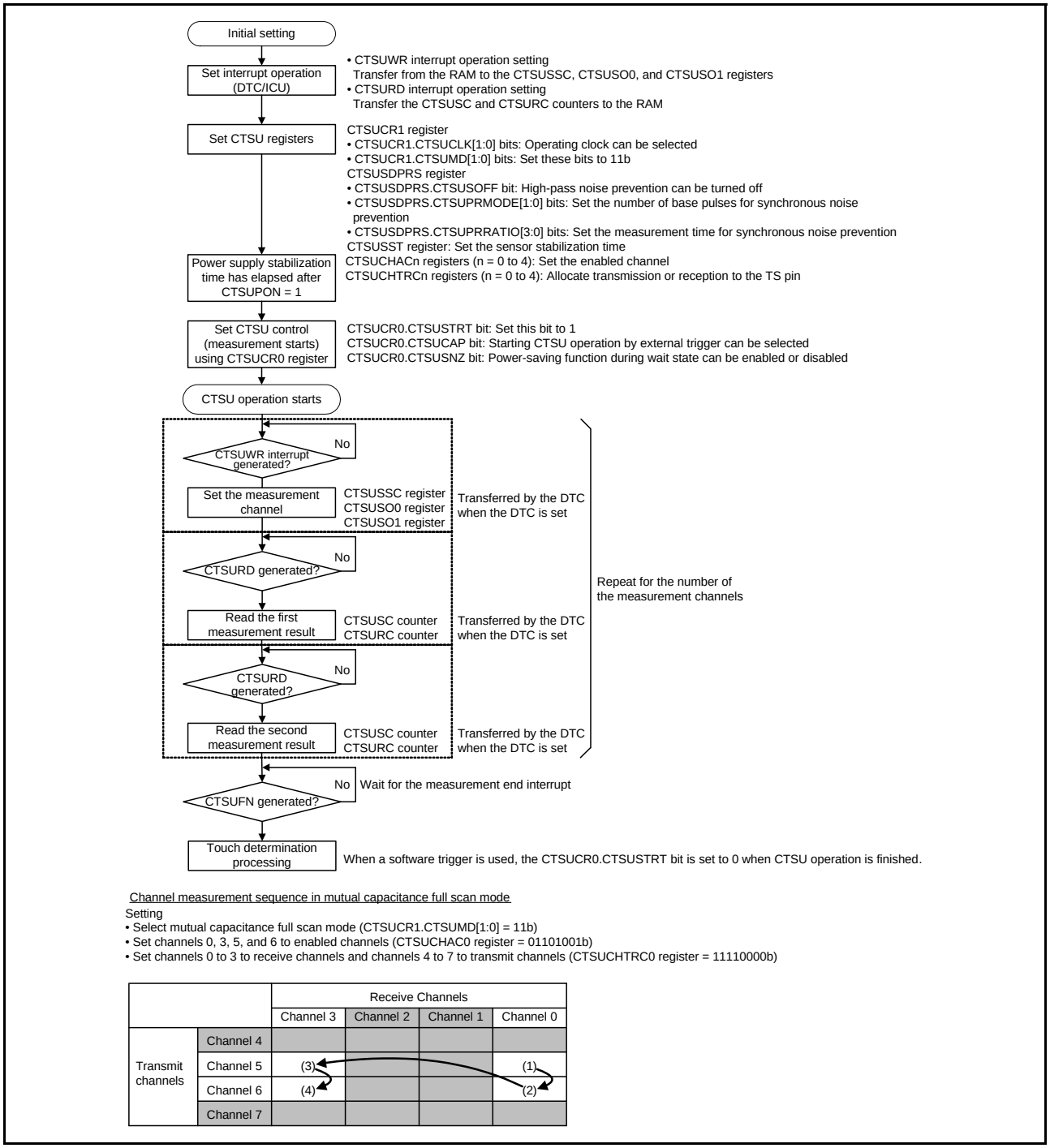



**Figure 43.16 Software Flowchart and Operation Example of Mutual Capacitance Full Scan Mode**

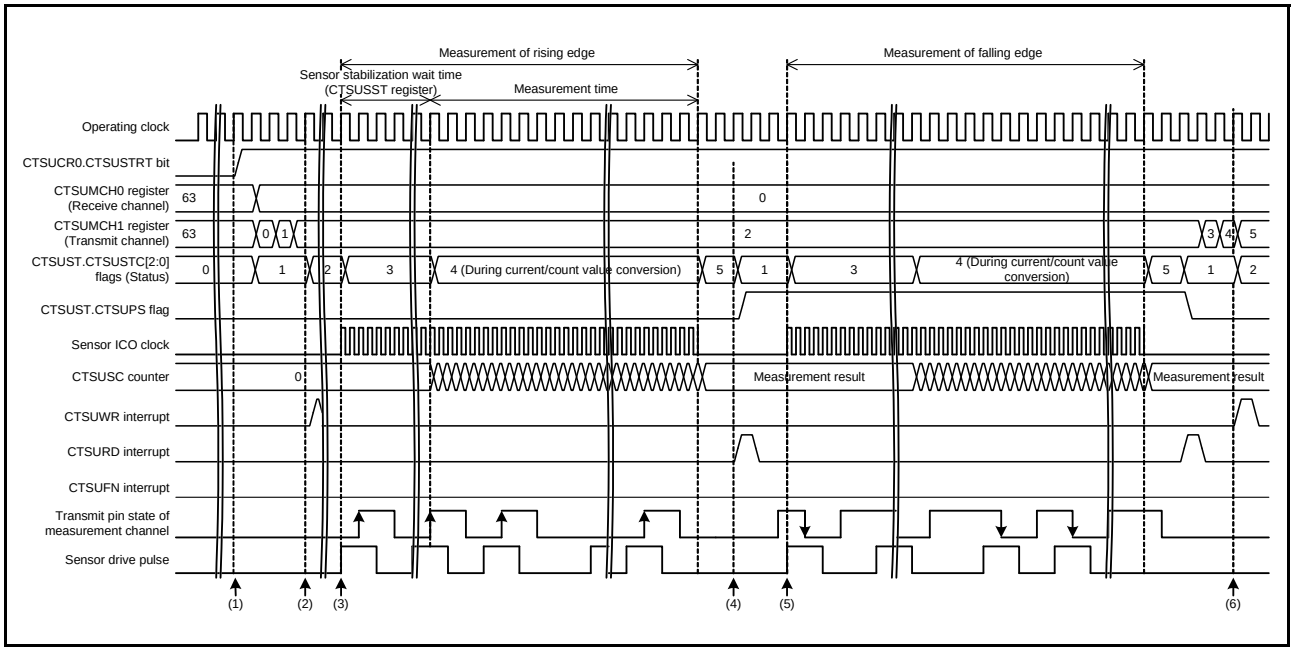



**Figure 43.17 Timing Chart of Mutual Capacitance Full Scan Mode (Measurement Start Condition is Software Trigger)**

The following describes operation shown in the timing chart in Figure 43.17.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
At the same time, a pulse which is handled as the rising edge is output to the transmit pin on the measurement channel during the high-level period of the sensor drive pulse.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) The same channel is measured by outputting a pulse that is handled as the falling edge during the high-level period of the sensor drive pulse.

(6) After the same channel is measured twice, a channel to be measured next is determined and measured in the similar way.

(7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

The mutual capacitance measurement status flag (CTSUST.CTSUPS flag) is changed when Status 5 transitions to Status 1.

Table 43.8 lists the touch pin states in mutual capacitance full scan mode.

**Table 43.8 Touch Pin States in Mutual Capacitance Full Scan Mode**

| | Touch Pin of Receive Channel | | Touch Pin of Transmit Channel | | |
|---|---|---|---|---|---|
| Status | Measurement Channel | Non-Measurement Channel | Measurement Channel | Non-Measurement Channel | Remarks |
| 0 | Low | Low | Low | Low | — |
| 1 | Low | Low | Low/High | Low | — |
| 2 | Low | Low | Low | Low | — |
| 3 | Pulse | Low | Pulse | Low | Pulse of the phase same as that of the receive channel at the first measurement<br>Pulse of the phase opposite to that of the receive channel at the second measurement |
| 4 | Pulse | Low | Pulse | Low | — |
| 5 | Low | Low | Low | Low | — |

### 43.3.3 Items Common to Multiple Modes

#### 43.3.3.1 Sensor Stabilization Wait Time and Measurement Time

Figure 43.18 shows the timing chart of the sensor stabilization wait time and measurement time.

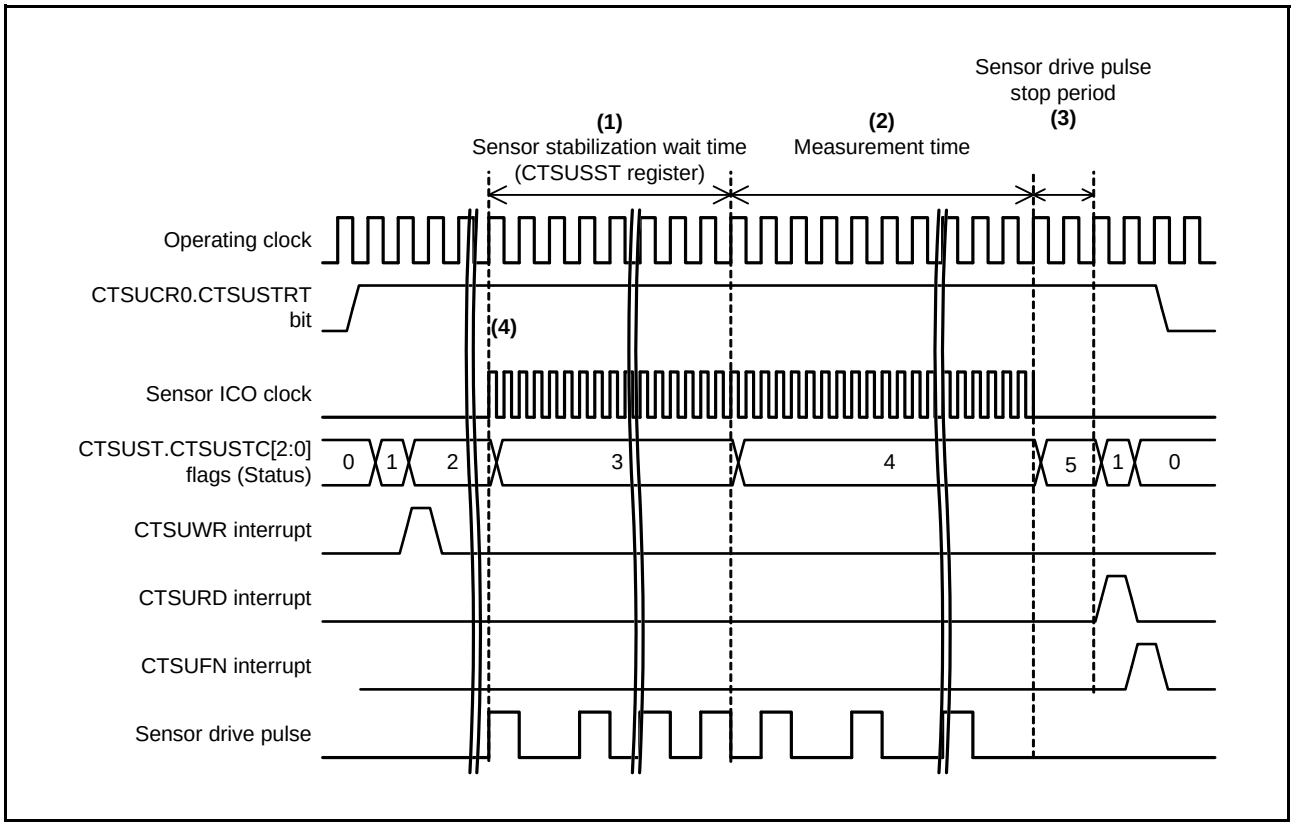



**Figure 43.18 Sensor Stabilization Wait Time and Measurement Time**

(1) In response to the CTSUWR interrupt request, output of the sensor drive pulse is started by write access to the CTSUSO1 register. Then, wait for the stabilization time set in the CTSUSST register.

(2) When the sensor stabilization time has elapsed and the CTSUST.CTSUDTSR flag is set to 0, measurement is started at transition to Status 4. The measurement time is determined by setting the base clock cycle and the CTSUSDPRS.CTSUPRMODE[1:0], CTSUPRRATIO[3:0], and CTSUSO0.CTSUSNUM[5:0] bits. When the measurement time has elapsed, measurement of the corresponding channel is finished.

(3) After the measurement time has elapsed, the status transitions to Status 1 after two operating clock cycles and a CTSURD interrupt is generated, so read the data from the CTSUSC and CTSURC counters.
At this time, the sensor drive pulse is output at the low level. When measurement of all specified channels is completed, the CTSUCR0.CTSUSTRT bit becomes 0.

(4) The sensor ICO clock oscillates while the CTSUSTC[2:0] flags are 011b (Status 3) or 100b (Status 4).

### 43.3.3.2 Interrupts

There are three types of interrupts for the CTSU:

- Write request interrupt for setting registers for each channel (CTSUWR)
- Measurement data transfer request interrupt (CTSURD)
- Measurement end interrupt (CTSUFN)

(1) Write request interrupt for setting registers for each channel (CTSUWR)

Store the setting data for each measurement channel in the RAM, and set the DTC or ICU transfer corresponding to the CTSUWR interrupt in advance. The CTSUWR interrupt is output when Status 1 transitions to Status 2. Write the setting data of the corresponding channel from the RAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers (Figure 43.19). Since write access to the CTSUSO1 register controls a transition to the next status, be sure to set this register last.

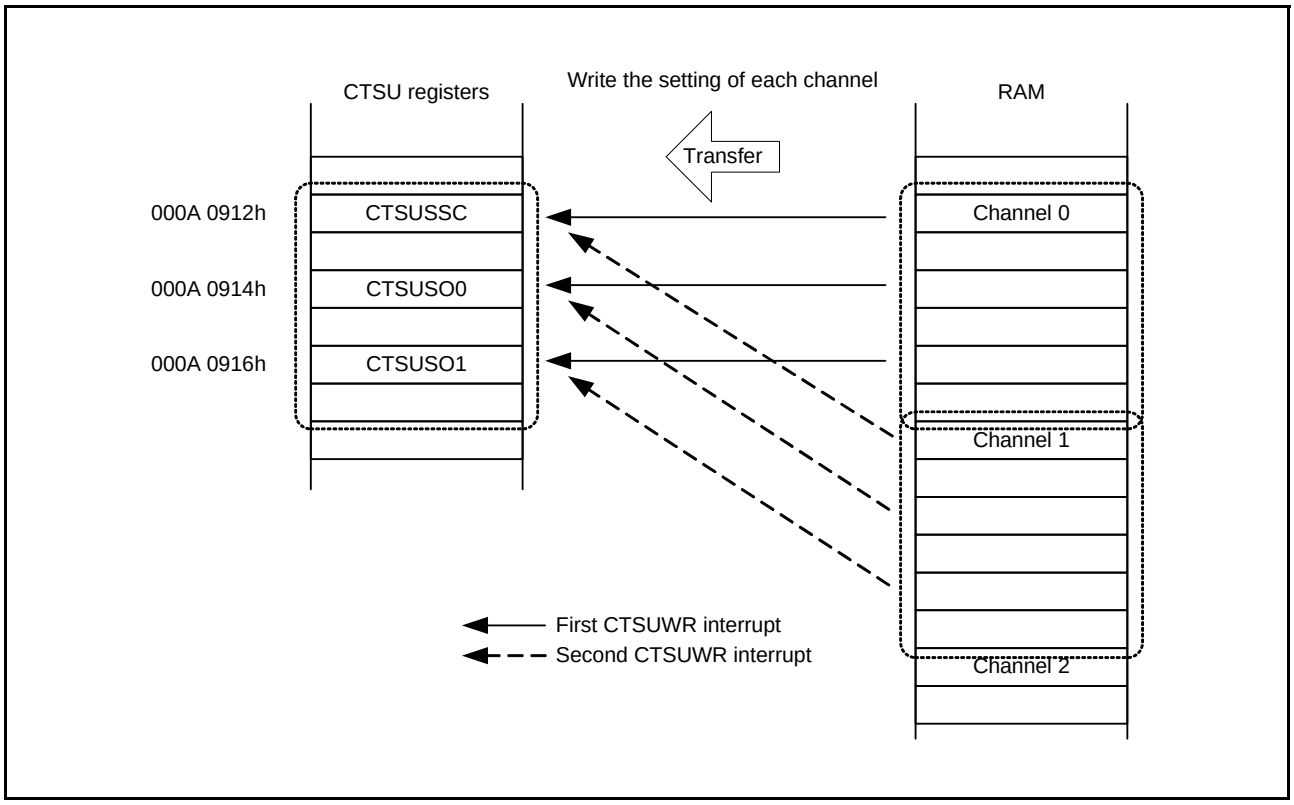



**Figure 43.19 Example of DTC Transfer Operation Using CTSUWR Interrupt**

The registers (CTSUSSC, CTSUSO0, and CTSUSO1 registers) to be set are allocated at sequential addresses. Set the operation at interrupt generation as shown below:

- Transfer destination address: Address of the CTSUSSC register
- Handling at the transfer destination address: Transfer 2-byte data three times by a single interrupt. (The address of the start byte is fixed.)
- Transfer source address: CTSUSSC register data storage address for the minimum channel in the setting data stored in the RAM
- Handling at the transfer source address: Transfer 2-byte data three times by a single interrupt. (The address of the first byte is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

(2) Measurement data transfer request interrupt (CTSURD)

Set DTC or ICU transfer corresponding to the CTSURD interrupt in advance. The CTSURD interrupt is output when Status 5 transitions to Status 1. Read the measurement result from the CTSUSC and CTSURC counters (Figure 43.20).

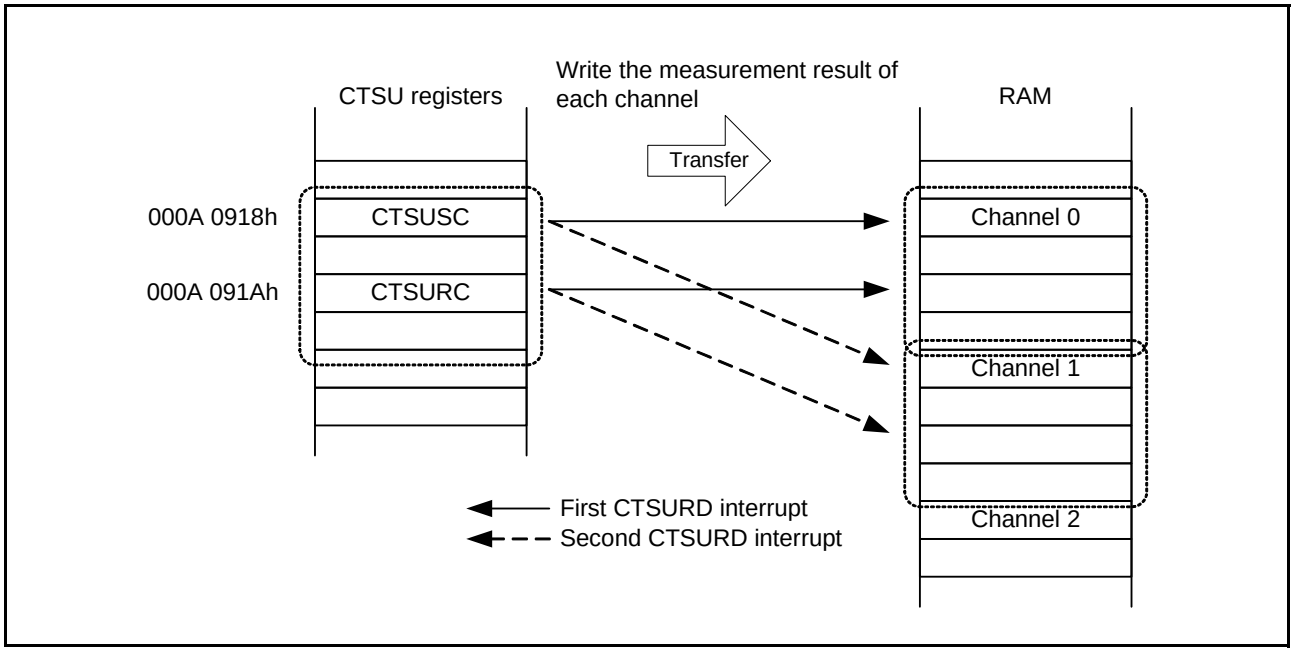



**Figure 43.20 Example of DTC Transfer Operation Using CTSURD Interrupt**

The measurement result registers (CTSUSC and CTSURC counters) used as transfer sources are allocated at sequential addresses. Set the operation at interrupt generation as shown below:

- Transfer source address: Address of the CTSUSC counter
- Handling at the transfer source address: Transfer 2-byte data twice by a single interrupt. (The start address is fixed.)
- Transfer destination address: CTSUSC counter data storage address for the minimum channel in the setting data stored in the RAM.
- Handling at the transfer destination address: Transfer 2-byte data twice by a single interrupt. (The start address is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

(3) Measurement end interrupt (CTSUFN)

When all channels are measured, an interrupt is generated when Status 1 transitions to Status 0. Use software to confirm the overflow flags (CTSUST.CTSUSOVF and CTSUROVF flags) and read the measurement results to determine whether or not the electrode is touched.

Interrupt requests are accepted or disabled in the interrupt control block.

## 43.4 Usage Notes

### 43.4.1 Measurement Result Data (CTSUSC and CTSURC Counters)

Read access during measurement is prohibited. If the measurement result data is accessed, an incorrect value may be read due to asynchronous operation.

### 43.4.2 Software Trigger

When 10b (PCLK/4) is selected by the CTSUCR1.CTSUCLK[1:0] bits, to restart measurement by writing 1 to the CTSUCR0.CTSUSTRT bit after measurement has been completed, wait for at least three cycles to elapse after an interrupt is generated, and then write to the CTSUCR0.CTSUSTRT bit.

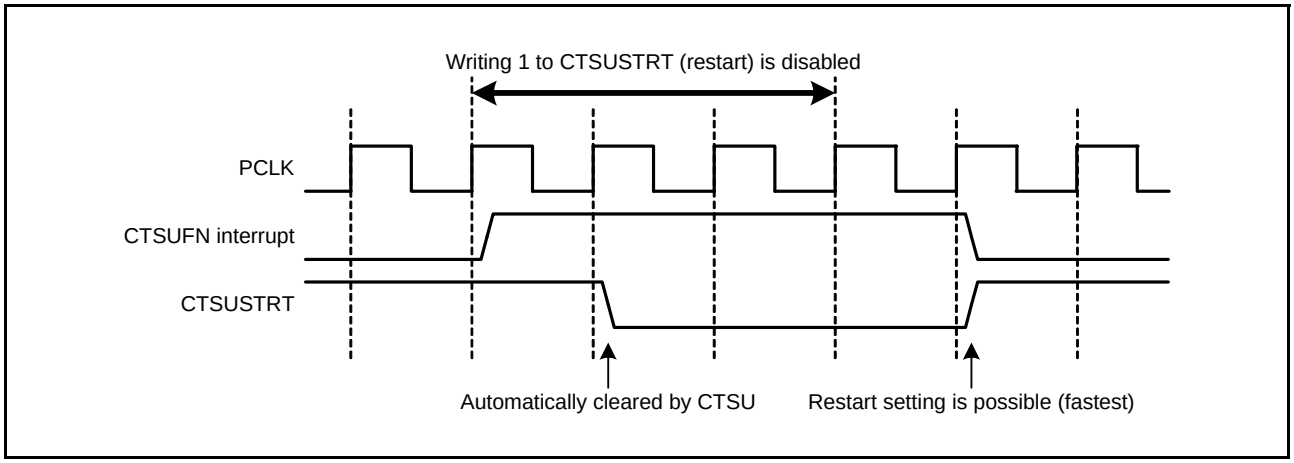



**Figure 43.21 Notes on Restarting Measurement**

### 43.4.3 External Trigger

- If an external trigger is input during the measurement time, measurement is not started. The next external event is enabled after one cycle of the operating clock when a CTSUFN interrupt is generated.
- To stop external trigger mode, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time (forced stop).

### 43.4.4 Notes on Forcibly Stopping Operation

To forcibly stop the current operation, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time. After this setting, the operation is stopped and the internal control registers are initialized.
When the CTSUCR0.CTSUINIT bit is used for initialization, the following registers are initialized in addition to the initialization of the internal measurement state.

- CTSUMCH0 register
- CTSUMCH1 register
- CTSUST register
- CTSUSC counter
- CTSURC counter

If operation is forcibly stopped, an interrupt request may be generated depending on the internal state. After operation is forcibly stopped, perform the processing for stopping/disabling the DTC or ICU.

If DTC transfer is stopped in the mounted system for some reason, also perform the processing for forcibly stopping and initializing the CTSU.

### 43.4.5 TSCAP Pin

The TSCAP pin requires an external decoupling capacitor to stabilize CTSU internal voltage. The traces between the TSCAP pin and the capacitor, and the capacitor and ground should be as short and wide as physically possible.
The capacitor connected to the TSCAP pin should be fully discharged using I/O port control to output a low level, before turning on the switch (CTSUCR1.CTSUCSW bit = 1) to establish a connection.

### 43.4.6 Notes during Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1)

During measurement operation (CTSUCR0.CTSUSTRT bit = 1), do not use settings such as “stop the peripheral module clock” or “change the port settings related to the touch pins (TS and TSCAP pins)” in the higher layers of the system.
If control settings non-compliant to these restrictions are made, after operation is forcibly stopped (CTSUCR0.CTSUSTRT bit = 0 and CTSUCR0.CTSUINIT bit = 1), write 0 to the CTSUCR1.CTSUPON bit and 0 to the CTSUCR1.CTSUCSW bit at the same time, and set the CTSUCR0.CTSUSNZ bit to 0. Then, restart from the initial setting flow shown in Figure 43.9.

# 44. 12-Bit A/D Converter (S12ADE)

In this section, “PCLK” is used to refer to PCLKB.

## 44.1 Overview

This MCU incorporates one unit of a 12-bit successive approximation A/D converter. Up to 14 channel analog inputs, temperature sensor output, and internal reference voltage are selectable for conversion.
The 12-bit A/D converter converts a maximum of 14 selected channels of analog inputs, temperature sensor output, and internal reference voltage, which have been selected, into a 12-bit digital value through successive approximation.
The A/D converter has three operating modes: single scan mode in which the analog inputs of up to 14 arbitrarily selected channels are converted only once in ascending channel order; and continuous scan mode in which the analog inputs of up to 14 arbitrarily selected channels are continuously converted in ascending channel order; and group scan mode in which up to 14 channels of the analog inputs are arbitrarily divided into two groups (group A and group B) and converted in ascending channel order in each group.
In group scan mode, the conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. When group-A priority control is selected along with operation as described above, if a request to start scanning for group A is received during A/D conversion for group B, the conversion operation for group B is discontinued and the conversion for group A starts, which is given priority.
In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).
Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.
It is prohibited to simultaneously select both temperature sensor output and internal reference voltage. Perform A/D conversion independently for the temperature sensor output or the internal reference voltage.
The external pin input (VREFH0) or the analog reference voltage (AVCC0) is selectable as the reference voltage on the high-potential side. The external pin input (VREFL0) or the analog reference voltage (AVSS0) is selectable as the reference voltage on the low-potential side.
This IP has a compare function (window A and window B). This function is used to specify the high-side reference value and low-side reference value for window A and window B, respectively. When the A/D-converted value of the selected channel meets the comparison conditions, the ELC event (S12ADWMELC/S12ADWUMELC) is output according to the event conditions (A or B, A and B, A exor B). Furthermore, the comparator operation to compare the A/D-converted value with the low-side reference value is also enabled.
The A/D data storage buffer is a ring buffer consisting of 16 buffers to sequentially store A/D converted data.
Table 44.1 lists the specifications of the 12-bit A/D converter and Table 44.2 lists the functions of the 12-bit A/D converter. Figure 44.1 shows a block diagram of the 12-bit A/D converter.

**Table 44.1 Specifications of 12-Bit A/D Converter (1/2)**

| Item | Description |
|---|---|
| Number of units | One unit |
| Input channels | Up to 14 channels |
| Extended analog function | Temperature sensor output, internal reference voltage |
| A/D conversion method | Successive approximation method |
| Resolution | 12 bits |
| Conversion time | 0.83 µs per channel<br>(when A/D conversion clock ADCLK = 54 MHz) |
| A/D conversion clock | Peripheral module clock PCLK*1 and A/D conversion clock ADCLK*1 can be set so that the frequency ratio should be one of the following.<br>PCLK to ADCLK frequency ratio = 1:1, 1:2, 2:1, 4:1, 8:1<br>ADCLK is set using the clock generation circuit. |
| Data registers | • 14 registers for analog input, 1 for A/D-converted data duplication in double trigger mode<br>• One register for temperature sensor output<br>• One register for internal reference voltage<br>• One register for self-diagnosis<br>• The results of A/D conversion are stored in 12-bit A/D data registers.<br>• 12-bit accuracy output for the results of A/D conversion<br>• The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits*2 in the A/D data registers in A/D-converted value addition mode.<br>• Double trigger mode (selectable in single scan and group scan modes):<br>The first piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece is stored in the duplication register. |
| Operating modes | • Single scan mode:<br>A/D conversion is performed only once on the analog inputs of up to 14 channels arbitrarily selected.<br>A/D conversion is performed only once on the temperature sensor output.<br>A/D conversion is performed only once on the internal reference voltage.<br>• Continuous scan mode:<br>A/D conversion is performed repeatedly on the analog inputs of up to 14 channels arbitrarily selected.<br>• Group scan mode:<br>Analog inputs of up to 14 channels arbitrarily selected, are divided into group A and group B, and A/D conversion of the analog input selected on a group basis is performed only once.<br>The conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently.<br>• Group scan mode (when group A is given priority):<br>If a group A trigger is input during A/D conversion on group B, the A/D conversion on group B is stopped and A/D conversion is performed on group A.<br>Restart (rescan) of A/D conversion on group B after completion of A/D conversion on group A can be set. |
| Conditions for A/D conversion start | • Software trigger<br>• Synchronous trigger<br>Trigger by the multi-function timer pulse unit (MTU), the event link controller (ELC), or the 16-bit timer pulse unit (TPU).<br>• Asynchronous trigger<br>A/D conversion can be triggered by the external trigger ADTRG0# pin. |
| Functions | • Variable sampling state count<br>• Self-diagnosis of 12-bit A/D converter<br>• Selectable A/D-converted value addition mode or average mode<br>• Analog input disconnection detection function (discharge function/precharge function)<br>• Double trigger mode (duplication of A/D conversion data)<br>• Automatic clear function of A/D data registers<br>• Compare function (window A and window B)<br>• 16 ring buffers when the compare function is used |

**Table 44.1 Specifications of 12-Bit A/D Converter (2/2)**

| Item | Description |
|---|---|
| Interrupt sources | • In the modes except double trigger mode and group scan mode, A/D scan end interrupt request (S12ADI0) can be generated on completion of single scan.<br>• In double trigger mode, A/D scan end interrupt request (S12ADI0) can be generated on completion of double scan.<br>• In group scan mode, an A/D scan end interrupt request (S12ADI0) can be generated on completion of group A scan, whereas an A/D scan end interrupt request (GBADI) for group B can be generated on completion of group B scan.<br>• When double trigger mode is selected in group scan mode, A/D scan end interrupt request (S12ADI0) can be generated on completion of double scan of group A, whereas A/D scan end interrupt request (GBADI) specially for group B can be generated on completion of group B scan.<br>• The S12ADI0 and GBADI interrupts can activate the DMA controller (DMAC) and the data transfer controller (DTC). |
| Event link function | • An ELC event is generated on completion of scans other than group B scan in group scan mode.<br>• An ELC event is generated on completion of group B scan in group scan mode.<br>• An ELC event is generated on completion of all scans.<br>• Scan can be started by a trigger output by the ELC.<br>• An ELC event is generated according to the event conditions of the window compare function in single scan mode. |
| Low power consumption function | • Module stop state can be set.*3, *4 |

Note 1. The peripheral module clock PCLK frequency is set according to the setting of the SCKCR.PCKB[3:0] bits and the A/D conversion clock ADCLK frequency is set according to the setting of the SCKCR.PCKD[3:0] bits.
Note 2. The number of extended bits during addition differs depending on the addition count.
2-bit extension: 1-time to 4-time conversion (addition zero to three times)
4-bit extension: 16-time conversion (addition 15 times)
Note 3. See section 11, Low Power Consumption for details.
Note 4. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

**Table 44.2 Functions of 12-Bit A/D Converter**

| Item | | | Pin Name, Abbreviation |
|---|---|---|---|
| Analog input channels | | | AN000 to AN007, AN016 to AN020, AN027, temperature sensor output, internal reference voltage |
| Conditions for A/D conversion start | Software | Software trigger | Enabled |
| | Asynchronous trigger | ADTRG0# | Enabled |
| | Synchronous trigger | Compare match/input capture from MTU0.TGRA | TRG0AN |
| | | Compare match/input capture from MTU0.TGRB | TRG0BN |
| | | Compare match/input capture from MTU0 to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode | TRGAN |
| | | Compare match from MTU0.TGRE | TRG0EN |
| | | Compare match from MTU0.TGRF | TRG0FN |
| | | Compare match between MTU4.TADCORA and MTU4.TCNT (interrupt skipping function) | TRG4AN |
| | | Compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function) | TRG4BN |
| | | Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function) | TRG4ABN |
| | | TGRA compare match/input capture from TPU0 to TPU4 or TGRA compare match/input capture from TPU0 | TRGAN1 TRG4ABN1 |
| | | ELC trigger | Enabled |
| Interrupt | | | S12ADI0, GBADI interrupt |
| Setting of the module stop function*1 | | | MSTPCRA. MSTPA17 bit |

Note 1. See section 11, Low Power Consumption for details.

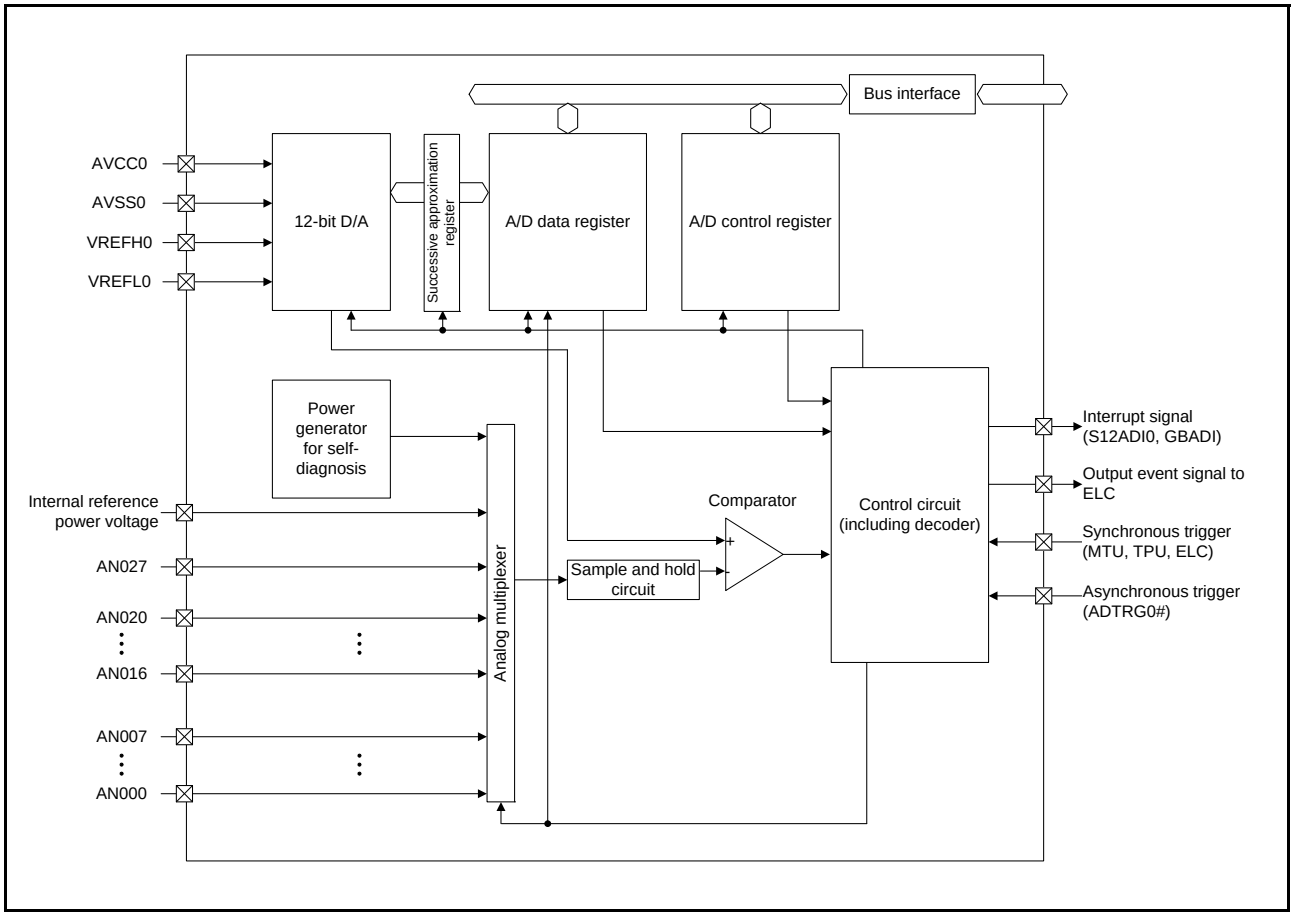



**Figure 44.1 Block Diagram of 12-Bit A/D Converter**

Table 44.3 lists the input pins of the 12-bit A/D converter.

**Table 44.3 Pin Configuration of 12-Bit A/D Converter**

| Pin Name | I/O | Function |
|---|---|---|
| AVCC0 | Input | Analog block power supply pin |
| AVSS0 | Input | Analog block ground pin |
| VREFH0 | Input | Reference power supply pin |
| VREFL0 | Input | Reference power supply ground pin |
| AN000 to AN007, AN016 to AN020, AN027 | Input | Analog input pins 0 to 7, analog input pins 16 to 20 and 27 |
| ADTRG0# | Input | External trigger input pin for starting A/D conversion |

## 44.2 Register Descriptions

### 44.2.1 A/D Data Registers y (ADDRy) (y = 0 to 7, 16 to 20, 27), A/D Data Duplication Register (ADDBLDR), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR)

Address(es): S12AD.ADDR0 0008 9020h, S12AD.ADDR1 0008 9022h, S12AD.ADDR2 0008 9024h, S12AD.ADDR3 0008 9026h, S12AD.ADDR4 0008 9028h, S12AD.ADDR5 0008 902Ah, S12AD.ADDR6 0008 902Ch, S12AD.ADDR7 0008 902Eh, S12AD.ADDR16 0008 9040h, S12AD.ADDR17 0008 9042h, S12AD.ADDR18 0008 9044h, S12AD.ADDR19 0008 9046h, S12AD.ADDR20 0008 9048h, S12AD.ADDR27 0008 9056h,S12AD.ADDBLDR 0008 9018h, S12AD.ADTSDR 0008 901Ah, S12AD.ADOCDR 0008 901Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ADDRy (y = 0 to 7, 16 to 20, 27) are 16-bit read-only registers which store the A/D conversion results.
ADDBLDR is a 16-bit read-only register used in double trigger mode. ADDBLDR stores the results of A/D conversion when the conversion is started by the second trigger.
ADTSDR is a 16-bit read-only register that stores the A/D conversion results of the temperature sensor output.
ADOCDR is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage.
The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (addition once, twice, three, or 15 times)
- Settings of the average mode enable bit (ADADC.AVEE) (addition or average)

The data formats for each given condition are shown below.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Average Mode is Selected

- Flush-right format
  The mean value of the A/D-converted results of the same channel is stored in bits 11 to 0.
  Bits 15 to 12 are read as 0.
- Flush-left format
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 4.
  Bits 3 to 0 are read as 0.

A/D-converted value average mode can be set only when twice or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)

The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
Bits 1 and 0 are read as 0.

- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the A/D data register as 2-bit extended data of the conversion accuracy bits; when the conversion count is set to 16 times, the value added by the A/D conversion result is retained in the A/D data register as 4-bit extended data of the conversion accuracy bits. Even if A/D-converted value addition mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bits.

## 44.2.2 A/D Self-Diagnosis Data Register (ADRD)

Address(es): S12AD.ADRD 0008 901Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ADRD is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. In the ADRD register, the different formats are used depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see section 44.2.11, A/D Control Extended Register (ADCER).

The data formats for each given condition are shown below.

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. The self-diagnosis status is stored in bits 15 and 14.
  Bits 13 and 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. The self-diagnosis status is stored in bits 1 and 0.
  Bits 3 and 2 are read as 0.

**Table 44.4 Self-Diagnosis Status Description**

| Bits 15 and 14 for flush-right format setting<br>Bits 1 and 0 for flush-left format setting | Self-diagnosis status |
|---|---|
| 00b | Self-diagnosis has never been executed since power-on. |
| 01b | Self-diagnosis using the voltage of 0 V has been executed. |
| 10b | Self-diagnosis using the reference voltage × 1/2 has been executed. |
| 11b | Self-diagnosis using the reference voltage has been executed. |

Note: For details of self-diagnosis, see section 44.2.11, A/D Control Extended Register (ADCER).

## 44.2.3 A/D Control Register (ADCSR)

Address(es): S12AD.ADCSR 0008 9000h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADST | ADCS[1:0] | | ADIE | — | ADHSC | TRGE | EXTRG | DBLE | GBADIE | — | DBLANS[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | DBLANS[4:0] | Double Trigger Channel Select | These bits select one analog input channel for double triggered operation. The setting is only effective while double trigger mode is selected. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | GBADIE | Group B Scan End Interrupt Enable | 0: Disables GBADI interrupt generation upon group B scan completion.<br>1: Enables GBADI interrupt generation upon group B scan completion. | R/W |
| b7 | DBLE | Double Trigger Mode Select | 0: Deselects double trigger mode.<br>1: Selects double trigger mode. | R/W |
| b8 | EXTRG | Trigger Select *1 | 0: A/D conversion is started by synchronous trigger.<br>1: A/D conversion is started by asynchronous trigger. | R/W |
| b9 | TRGE | Trigger Start Enable | 0: Disables A/D conversion to be started by synchronous or asynchronous trigger.<br>1: Enables A/D conversion to be started by synchronous or asynchronous trigger. | R/W |
| b10 | ADHSC | A/D Conversion Select | 0: High-speed conversion<br>1: Low-current conversion | R/W |
| b11 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b12 | ADIE | Scan End Interrupt Enable | 0: Disables S12ADI0 interrupt generation upon scan completion.<br>1: Enables S12ADI0 interrupt generation upon scan completion. | R/W |
| b14, b13 | ADCS[1:0] | Scan Mode Select | b14 b13<br>0 0: Single scan mode<br>0 1: Group scan mode<br>1 0: Continuous scan mode<br>1 1: Setting prohibited | R/W |
| b15 | ADST | A/D Conversion Start | 0: Stops A/D conversion process.<br>1: Starts A/D conversion process. | R/W |

Note 1. Starting A/D conversion using an external pin (asynchronous trigger)
After a high-level signal is input to the external pin (ADTRG0#), write 1 to both the TRGE and EXTRG bits in ADCSR and change the signals of ADTRG0# to low. Thus the falling edge of ADTRG0# is detected and the scan conversion process is started. In this case, the pulse width of the low-level input must be at least 1.5 clock cycles of PCLK.

ADCSR sets double trigger mode, A/D conversion start trigger; enables/disables scan end interrupt; selects the scan mode; and starts or stops A/D conversion.

### DBLANS[4:0] Bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into the A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 44.5 shows selection of the channel for double triggered operation.

When double trigger mode is selected, channel selection using the ADANSA0 and ADANSA1 registers is invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.

When double trigger mode is used, do not select A/D conversion for the self-diagnosis function, temperature sensor output, and internal reference voltage (temperature sensor output and internal reference voltage can be selected for A/D conversion for group B in group scan mode). The DBLANS[4:0] bits should be set while the ADST bit is 0. They should

not be set simultaneously when 1 is written to the ADST bit.
To enter A/D-converted value addition/average mode while double trigger mode is set, the channel selected by the DBLANS[4:0] bits should be selected in the ADANSA0 and ADANSA1 registers.

**Table 44.5 Relationship between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels**

| DBLANS[4:0] | Duplication Channel | DBLANS[4:0] | Duplication Channel |
|---|---|---|---|
| 00000b | AN000 | 10000b | AN016 |
| 00001b | AN001 | 10001b | AN017 |
| 00010b | AN002 | 10010b | AN018 |
| 00011b | AN003 | 10011b | AN019 |
| 00100b | AN004 | 10100b | AN020 |
| 00101b | AN005 | 11011b | AN027 |
| 00110b | AN006 | | |
| 00111b | AN007 | | |

### GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt (GBADI) in group scan mode.

### DBLE Bit (Double Trigger Mode Select)

Double trigger mode has a function to store the resulting data of A/D conversion started by the first and second synchronous triggers into separate registers.
When double trigger mode is selected, channel selection using the ADANSA0 and ADANSA1 registers is invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits. Do not generate an asynchronous or software trigger.
The A/D conversion results started by the first trigger are stored into the A/D data register y and those started by the second trigger are stored into the A/D data duplication register. In this case, if the ADIE bit is set to 1, the interrupt is generated not upon completion of the first conversion but upon completion of the second conversion.
In continuous scan mode, double trigger mode should not be selected.
The DBLE bit should be set after the ADST bit has been set to 0.

### EXTRG Bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or the asynchronous trigger as the trigger for starting A/D conversion.

### TRGE Bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger.
This bit should be set to 1 in group scan mode.

### ADHSC Bit (A/D Conversion Select)

The ADHSC bit sets the operating mode of A/D conversion. When modifying this bit, set the 12-bit converter to the standby state. For the procedure for modifying the ADHSC bit, see section 44.8.10, ADHSC Bit Rewriting Procedure.

### ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the A/D scan end interrupt (S12ADI0) in scans except for group B scan in group scan mode.
With double trigger mode deselected, the S12ADI0 interrupt is generated after the first scan is completed if the ADIE bit is set to 1.
With double trigger mode selected, the S12ADI0 interrupt is generated after the second scan is completed if the ADIE bit

is set to 1 as long as the scan is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits.

### ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 14 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.
In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of a maximum of 14 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion is stopped even if scanning is in progress.
In group scan mode, A/D conversion is performed for the analog inputs (group A) of 14 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B) of a maximum of 14 channels selected with the ADANSB0 and ADANSB1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped.
When selecting group scan mode, different channels and triggers should be selected for group A and group B.
When selecting the temperature sensor output or internal reference voltage, select single scan mode, and deselect all the channels selected with the ADANSA0 and ADANSA1 registers before performing A/D conversion. When A/D conversion of the selected temperature sensor output or internal reference voltage is completed, A/D conversion is stopped.
The ADCS[1:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

### ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.
Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.
[Setting conditions]

- 1 is written by software.
- The synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- The synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[5:0] bits being set to 000000b.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group B trigger is detected and A/D conversion of group B is started.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and A/D conversion of group B is restarted.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and A/D conversion of group B is started.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the selected channels, the temperature sensor output, or the internal reference voltage is completed in single scan mode.

- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group A trigger is detected during group B A/D conversion and the scanning of group B is stopped.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and the scanning of group B started by a resumption trigger is completed.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) the ADGSPCR.GBRP bit is set to 1 and the scanning of group B by a trigger is completed.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) and ADGSPCR.GBRP = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

## 44.2.4 A/D Channel Select Register A0 (ADANSA0)

Address(es): S12AD.ADANSA0 0008 9004h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ANSA007 | ANSA006 | ANSA005 | ANSA004 | ANSA003 | ANSA002 | ANSA001 | ANSA000 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ANSA000 | A/D Conversion Channel Select | 0: AN000 to AN007 are not subjected to conversion.<br>1: AN000 to AN007 are subjected to conversion. | R/W |
| b1 | ANSA001 | | | R/W |
| b2 | ANSA002 | | | R/W |
| b3 | ANSA003 | | | R/W |
| b4 | ANSA004 | | | R/W |
| b5 | ANSA005 | | | R/W |
| b6 | ANSA006 | | | R/W |
| b7 | ANSA007 | | | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADANSA0 selects analog input channels for A/D conversion among AN000 to AN007. In group scan mode, this register selects group A channels.

**ANSA0n Bit (n = 00 to 07) (A/D Conversion Channel Select)**

The ANSA0n bit selects analog input channels for A/D conversion among AN000 to AN007. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN000 and the ANSA007 bit corresponds to AN007.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

When double trigger mode is selected, the channel selected by the ANSA0n bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA0n bit should be set while the ADCSR.ADST bit is 0.

## 44.2.5 A/D Channel Select Register A1 (ADANSA1)

Address(es): S12AD.ADANSA1 0008 9006h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | ANSA111 | — | — | — | — | — | — | ANSA104 | ANSA103 | ANSA102 | ANSA101 | ANSA100 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ANSA100 | A/D Conversion Channel Select | 0: AN016 to AN020 are not subjected to conversion.<br>1: AN016 to AN020 are subjected to conversion. | R/W |
| b1 | ANSA101 | | | R/W |
| b2 | ANSA102 | | | R/W |
| b3 | ANSA103 | | | R/W |
| b4 | ANSA104 | | | R/W |
| b10 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | ANSA111 | A/D Conversion Channel Select | 0: AN027 is not subjected to conversion.<br>1: AN027 is subjected to conversion. | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADANSA1 selects analog input channels for A/D conversion among AN016 to AN020 and AN027. In group scan mode, group A channels are to be selected.

**ANSA1n Bit (n = 00 to 04, 11) (A/D Conversion Channel Select)**

The ANSA1n bit (n = 00 to 04, 11) select analog input channels for A/D conversion among AN016 to AN020 and AN027. The channels to be selected and the number of channels can be arbitrarily set. The ANSA100 bit corresponds to AN016 and the ANSA111 bit corresponds to AN027.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

When double trigger mode is selected, the channel selected by the ANSA1n bit is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA1n bit should be set while the ADCSR.ADST bit is 0.

## 44.2.6 A/D Channel Select Register B0 (ADANSB0)

Address(es): S12AD.ADANSB0 0008 9014h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ANSB007 | ANSB006 | ANSB005 | ANSB004 | ANSB003 | ANSB002 | ANSB001 | ANSB000 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ANSB000 | A/D Conversion Channel Select | 0: AN000 to AN007 are not subjected to conversion.<br>1: AN000 to AN007 are subjected to conversion. | R/W |
| b1 | ANSB001 | | | R/W |
| b2 | ANSB002 | | | R/W |
| b3 | ANSB003 | | | R/W |
| b4 | ANSB004 | | | R/W |
| b5 | ANSB005 | | | R/W |
| b6 | ANSB006 | | | R/W |
| b7 | ANSB007 | | | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADANSB0 selects analog input channels for A/D conversion among AN000 to AN007 in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

**ANSB0n Bit (n = 00 to 07) (A/D Conversion Channel Select)**

The ANSB0n bit selects analog input channels for A/D conversion among AN000 to AN007 in group B when group scan mode is selected. The ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN000 and the ANSB007 bit corresponds to AN007.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

The ANSB0n bit should be set while the ADCSR.ADST bit is 0.

## 44.2.7 A/D Channel Select Register B1 (ADANSB1)

Address(es): S12AD.ADANSB1 0008 9016h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | ANSB111 | — | — | — | — | — | — | ANSB104 | ANSB103 | ANSB102 | ANSB101 | ANSB100 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ANSB100 | A/D Conversion Channel Select | 0: AN016 to AN020 are not subjected to conversion.<br>1: AN016 to AN020 are subjected to conversion. | R/W |
| b1 | ANSB101 | | | R/W |
| b2 | ANSB102 | | | R/W |
| b3 | ANSB103 | | | R/W |
| b4 | ANSB104 | | | R/W |
| b10 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | ANSB111 | A/D Conversion Channel Select | 0: AN027 is not subjected to conversion.<br>1: AN027 is subjected to conversion. | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADANSB1 selects analog input channels for A/D conversion among AN016 to AN020 and AN027 in group B when group scan mode is selected. The ADANSB1 register is not used in any scan mode other than group scan mode.

**ANSB1n Bit (n = 00 to 04, 11) (A/D Conversion Channel Select)**

The ANSB1n bit selects analog input channels for A/D conversion among AN016 to AN020 and AN027 in group B when group scan mode is selected. The ADANSB1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB100 bit corresponds to AN016 and the ANSB111 bit corresponds to AN027.

When performing A/D conversion of the temperature sensor output or internal reference voltage, do not select analog input channels. The setting value of this register should be 0000h.

The ANSB1n bit should be set while the ADCSR.ADST bit is 0.

## 44.2.8 A/D-Converted Value Addition/Average Function Select Register 0 (ADADS0)

Address(es): S12AD.ADADS0 0008 9008h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ADS007 | ADS006 | ADS005 | ADS004 | ADS003 | ADS002 | ADS001 | ADS000 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ADS000 | A/D-Converted Value Addition/ Average Channel Select | 0: A/D-converted value addition/average mode for AN000 to AN007 is not selected.<br>1: A/D-converted value addition/average mode for AN000 to AN007 is selected. | R/W |
| b1 | ADS001 | | | R/W |
| b2 | ADS002 | | | R/W |
| b3 | ADS003 | | | R/W |
| b4 | ADS004 | | | R/W |
| b5 | ADS005 | | | R/W |
| b6 | ADS006 | | | R/W |
| b7 | ADS007 | | | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADADS0 selects the channels 0 to 7 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

**ADS0n Bit (n = 00 to 07) (A/D-Converted Value Addition/Average Channel Select)**

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the ADANSA0.ANSA0n bit (n = 00 to 07) or ADCSR.DBLANS[4:0] bits and ADANSB0.ANSB0n bit (n = 00 to 07) is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.

The ADS0n bit should be set while the ADCSR.ADST bit is 0.

## 44.2.9 A/D-Converted Value Addition/Average Function Select Register 1 (ADADS1)

Address(es): S12AD.ADADS1 0008 900Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | ADS111 | — | — | — | — | — | — | ADS104 | ADS103 | ADS102 | ADS101 | ADS100 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ADS100 | A/D-Converted Value Addition/ Average Channel Select | 0: A/D-converted value addition/average mode for AN016 to AN020 is not selected.<br>1: A/D-converted value addition/average mode for AN016 to AN020 is selected. | R/W |
| b1 | ADS101 | | | R/W |
| b2 | ADS102 | | | R/W |
| b3 | ADS103 | | | R/W |
| b4 | ADS104 | | | R/W |
| b10 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | ADS111 | A/D-Converted Value Addition/ Average Channel Select | 0: A/D-converted value addition/average mode for AN027 is not selected.<br>1: A/D-converted value addition/average mode for AN027 is selected. | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADADS1 selects the channels 16 to 20 and 27 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

**ADS1n Bit (n = 00 to 04, 11) (A/D-Converted Value Addition/Average Channel Select)**

When the ADS1n bit of the number that is the same as that of A/D-converted channel selected by the ADANSA1.ANSA1n bit (n = 00 to 04, 11) or ADCSR.DBLANS[4:0] bits and ADANSB1.ANSB1n bit (n = 00 to 04, 11) is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.

The ADS1n bit should be set while the ADCSR.ADST bit is 0.

Figure 44.2 shows a scanning operation sequence in which both the ADS002 and ADS006 bits are set to 1.

It is assumed that addition mode is selected (ADADC.AVEE = 0), the addition count is set to three times (ADADC.ADC[2:0] = 011b), and the channels AN000 to AN007 are selected (ADANSA0.ANSA0n = FFh) in continuous scan mode (ADCSR.ADCS[1:0] = 10b). The conversion process begins with AN000. The AN002 conversion is performed successively four times (addition three times), and the added (integrated) value is stored in A/D data register 2. After that the AN003 conversion is started. The AN006 conversion is performed successively 4 times and the added (integrated) value is stored in A/D data register 6. After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000.

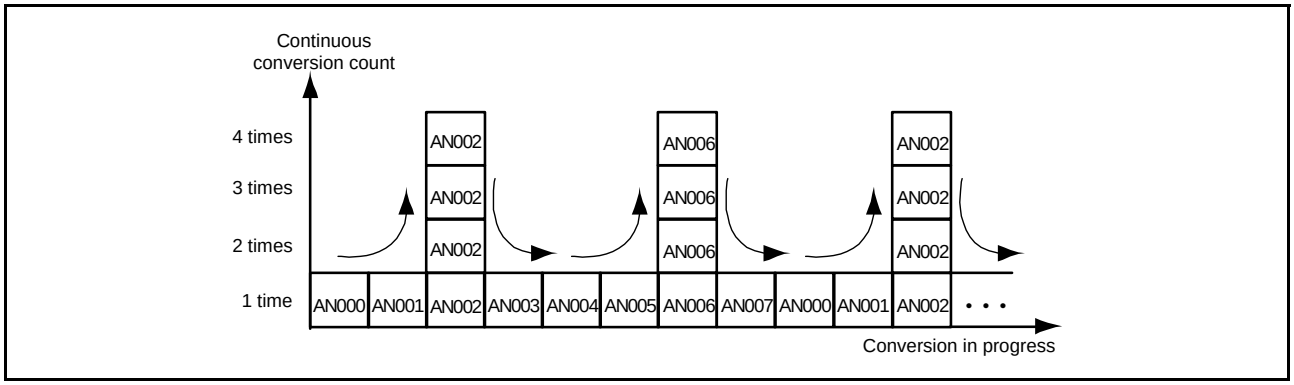



**Figure 44.2 Scan Conversion Sequence with ADADC.ADC[2:0] = 011b, ADS002 = 1, and ADS006 = 1**

## 44.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): S12AD.ADADC 0008 900Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | AVEE | — | — | — | — | ADC[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | ADC[2:0] | Addition Count Select | b2 b0<br>0 0 0: 1-time conversion (no addition; same as normal conversion)<br>0 0 1: 2-time conversion (addition once)<br>0 1 0: 3-time conversion (addition twice)*1<br>0 1 1: 4-time conversion (addition three times)<br>1 0 1: 16-time conversion (addition 15 times)*1<br>Settings other than above are prohibited. | R/W |
| b6 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | AVEE | Average Mode Enable | 0: Addition mode is selected.<br>1: Average mode is selected. | R/W |

Note 1. The AVEE bit is enabled only when 2-time or 4-time conversion is selected. When average mode is selected (ADADC.AVEE bit = 1), do not set 3-time conversion (ADADC.ADC[2:0] = 010b) nor 16-time conversion (ADADC.ADC[2:0] = 101b).

ADADC sets the number of times addition is to proceed for the temperature sensor output, the internal reference voltage, and channels selected as being in A/D-converted value addition or average mode, and to select either addition or average mode for them.

### ADC[2:0] Bits (Addition Count Select)

The ADC[2:0] bits set the number of times addition is to proceed as a common value for the temperature sensor output, the internal reference voltage, and channels for which A/D-converted value addition or average mode is selected, including those channels selected in double trigger mode (by the ADCSR.DBLANS[4:0] bits).
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b).
The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.

### AVEE Bit (Average Mode Enable)

The AVEE bit selects addition or average mode for the temperature sensor output, the internal reference voltage, and the channels selected for which the addition or average mode of A/D conversion is selected, including those channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits).
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b). The mean value of 1-time, 3-time, and 16-time conversion cannot be obtained.
The AVEE bit should be set while the ADCSR.ADST bit is 0.

## 44.2.11 A/D Control Extended Register (ADCER)

Address(es): S12AD.ADCER 0008 900Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADRFMT | — | — | — | DIAGM | DIAGLD | DIAGVAL[1:0] | | — | — | ACE | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b5 | ACE | A/D Data Register Automatic Clearing Enable | 0: Disables automatic clearing.<br>1: Enables automatic clearing. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b9, b8 | DIAGVAL[1:0] | Self-Diagnosis Conversion Voltage Select | b9 b8<br>0 0: Setting prohibited in self-diagnosis voltage fixed mode<br>0 1: Uses the voltage of 0 V for self-diagnosis.<br>1 0: Uses the reference voltage x 1/2 for self-diagnosis.*1<br>1 1: Uses the reference voltage for self-diagnosis.*1 | R/W |
| b10 | DIAGLD | Self-Diagnosis Mode Select | 0: Rotation mode for self-diagnosis voltage<br>1: Fixed mode for self-diagnosis voltage | R/W |
| b11 | DIAGM | Self-Diagnosis Enable | 0: Disables self-diagnosis of 12-bit A/D converter.<br>1: Enables self-diagnosis of 12-bit A/D converter. | R/W |
| b14 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 | ADRFMT | A/D Data Register Format Select | 0: Flush-right is selected for the A/D data register format.<br>1: Flush-left is selected for the A/D data register format. | R/W |

Note 1. The reference voltage refers to the voltage on the pin selected in the ADHVREFCNT register.

ADCER sets self-diagnosis mode, format of A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

### ACE Bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all “0”) of ADDRy, ADRD, ADDBLDR, ADTSDR, or ADOCDR after any of these registers have been read by the CPU, DTC, or DMACA. Automatic clearing of the A/D data register is enabled to detect a failure which has not been updated in the A/D data register.

### DIAGVAL[1:0] Bits (Self-Diagnosis Conversion Voltage Select)

These bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the ADCER.DIAGLD bit.

Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.

### DIAGLD Bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference voltage × 1/2, and the reference voltage are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

**DIAGM Bit (Self-Diagnosis Enable)**

The DIAGM bit enables or disables self-diagnosis.

Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the internally generated voltage values 0, the reference voltage × 1/2, and the reference voltage is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed in groups A and B.

The DIAGM bit should be set while the ADCSR.ADST bit is 0.

**ADRFMT Bit (A/D Data Register Format Select)**

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADCMPDR0, ADCMPDR1, ADWINLLB, or ADWINULB.

The ADRFMT bit should be set while the ADCSR.ADST bit is 0.

For details on the format of each data register, see section 44.2.1, A/D Data Registers y (ADDRy) (y = 0 to 7, 16 to 20, 27), A/D Data Duplication Register (ADDBLDR), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR), section 44.2.2, A/D Self-Diagnosis Data Register (ADRD), section 44.2.25, A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0), section 44.2.26, A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1), section 44.2.33, A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB), and section 44.2.34, A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB).

## 44.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): S12AD.ADSTRGR 0008 9010h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | TRSA[5:0] | | | | | | — | — | TRSB[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | TRSB[5:0] | A/D Conversion Start Trigger Select for Group B | Select the A/D conversion start trigger for group B in group scan mode. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b13 to b8 | TRSA[5:0] | A/D Conversion Start Trigger Select | Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected. | R/W |
| b15, b14 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADSTRGR selects the A/D conversion start trigger.

**TRSB[5:0] Bits (A/D Conversion Start Trigger Select for Group B)**

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. Therefore, the TRSB[5:0] bits should be set to the value other than 000000b and the ADCSR.TRGE bit should be set to 1 in group scan mode.
When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh.
Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time ($t_{SCAN}$). If the issuance period is less than $t_{SCAN}$, A/D conversion by the trigger may have no effect.
When the trigger from the module operated in 54 MHz (MTU) is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 44.3.6, Analog Input Sampling Time and Scan Conversion Time for details.
Table 44.6 lists the A/D conversion startup sources selected by the TRSB[5:0] bits.

**TRSA[5:0] Bits (A/D Conversion Start Trigger Select)**

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group scan mode or double trigger mode, software trigger and asynchronous trigger cannot be used.

- When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.
- When using the asynchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, and the TRSA[5:0] bits.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time ($t_{SCAN}$). If the issuance period is less than $t_{SCAN}$, A/D conversion by a trigger may have no effect. When the trigger from the module operated in 54 MHz (MTU) is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 44.3.6, Analog Input Sampling Time and Scan Conversion Time for details.

Table 44.7 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

**Table 44.6 Selection of A/D Activation Sources by the TRSB[5:0] Bits**

| Module | Source | Remarks | TRSB[5] | TRSB[4] | TRSB[3] | TRSB[2] | TRSB[1] | TRSB[0] |
|---|---|---|---|---|---|---|---|---|
| Trigger source deselection state | | | 1 | 1 | 1 | 1 | 1 | 1 |
| MTU | TRG0AN | Compare match/input capture from MTU0.TGRA | 0 | 0 | 0 | 0 | 0 | 1 |
| | TRG0BN | Compare match/input capture from MTU0.TGRB | 0 | 0 | 0 | 0 | 1 | 0 |
| | TRGAN | Compare match/input capture from MTU0 to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode | 0 | 0 | 0 | 0 | 1 | 1 |
| | TRG0EN | Compare match from MTU0.TGRE | 0 | 0 | 0 | 1 | 0 | 0 |
| | TRG0FN | Compare match from MTU0.TGRF | 0 | 0 | 0 | 1 | 0 | 1 |
| | TRG4AN | Compare match between MTU4.TADCORA and MTU4.TCNT (interrupt skipping function) | 0 | 0 | 0 | 1 | 1 | 0 |
| | TRG4BN | Compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function) | 0 | 0 | 0 | 1 | 1 | 1 |
| | TRG4ABN | Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function) | 0 | 0 | 1 | 0 | 0 | 0 |
| TPU | TRGAN1 | TGRA compare match/input capture from TPU0 to TPU4 | 0 | 0 | 1 | 1 | 0 | 1 |
| | TRG4ABN1 | TGRA compare match/input capture from TPU0 | 0 | 0 | 1 | 1 | 1 | 0 |
| ELC | ELCTRG0 | | 0 | 0 | 1 | 0 | 0 | 1 |

**Table 44.7 Selection of A/D Activation Sources by the TRSA[5:0] Bits**

| Module | Source | Remarks | TRSA[5] | TRSA[4] | TRSA[3] | TRSA[2] | TRSA[1] | TRSA[0] |
|---|---|---|---|---|---|---|---|---|
| Trigger source deselection state | | | 1 | 1 | 1 | 1 | 1 | 1 |
| External pin | ADTRG0# | Input pin for the trigger | 0 | 0 | 0 | 0 | 0 | 0 |
| MTU | TRG0AN | Compare match/input capture from MTU0.TGRA | 0 | 0 | 0 | 0 | 0 | 1 |
| | TRG0BN | Compare match/input capture from MTU0.TGRB | 0 | 0 | 0 | 0 | 1 | 0 |
| | TRGAN | Compare match/input capture from MTU0 to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode | 0 | 0 | 0 | 0 | 1 | 1 |
| | TRG0EN | Compare match from MTU0.TGRE | 0 | 0 | 0 | 1 | 0 | 0 |
| | TRG0FN | Compare match from MTU0.TGRF | 0 | 0 | 0 | 1 | 0 | 1 |
| | TRG4AN | Compare match between MTU4.TADCORA and MTU4.TCNT (interrupt skipping function) | 0 | 0 | 0 | 1 | 1 | 0 |
| | TRG4BN | Compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function) | 0 | 0 | 0 | 1 | 1 | 1 |
| | TRG4ABN | Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT (interrupt skipping function) | 0 | 0 | 1 | 0 | 0 | 0 |
| TPU | TRGAN1 | TGRA compare match/input capture from TPU0 to TPU4 | 0 | 0 | 1 | 1 | 0 | 1 |
| | TRG4ABN1 | TGRA compare match/input capture from TPU0 | 0 | 0 | 1 | 1 | 1 | 0 |
| ELC | ELCTRG0 | | 0 | 0 | 1 | 0 | 0 | 1 |

## 44.2.13 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): S12AD.ADEXICR 0008 9012h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | OCSA | TSSA | — | — | — | — | — | — | OCSAD | TSSAD |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TSSAD | Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select | 0: Temperature sensor output A/D-converted value addition/average mode is not selected.<br>1: Temperature sensor output A/D-converted value addition/average mode is selected. | R/W |
| b1 | OCSAD | Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select | 0: Internal reference voltage A/D-converted value addition/average mode is not selected.<br>1: Internal reference voltage A/D-converted value addition/average mode is selected. | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | TSSA | Temperature Sensor Output A/D Conversion Select | 0: A/D conversion of temperature sensor output is not performed.<br>1: A/D conversion of temperature sensor output is performed. | R/W |
| b9 | OCSA | Internal Reference Voltage A/D Conversion Select | 0: A/D conversion of internal reference voltage is not performed.<br>1: A/D conversion of internal reference voltage is performed. | R/W |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADEXICR specifies the settings of A/D conversion of the temperature sensor output and internal reference voltage.

**TSSAD Bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)**

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D temperature sensor data register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is stored in the A/D temperature sensor data register (ADTSDR).
The TSSAD bit should be set while the ADCSR.ADST bit is 0.

**OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)**

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is stored in ADOCDR.
The OCSAD bit should be set while the ADCSR.ADST bit is 0.

**TSSA Bit (Temperature Sensor Output A/D Conversion Select)**

This bit selects A/D conversion of the temperature sensor output in single scan mode. When A/D conversion of the temperature sensor output is to be performed, all the bits in the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers and the ADCSR.DBLE and OCSA bits should all be set to 0 in single scan mode.
The TSSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the temperature sensor output, the ADDISCR.ADNDIS[4:0] bits should be automatically set to 0Fh to discharge the A/D converter before sampling. The sampling time should be 5 µs or longer.

Sampling starts after discharging is completed during A/D conversion of the temperature sensor output, an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

### OCSA Bit (Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage in single scan mode. When A/D conversion of the internal reference voltage is to be performed, set all the bits in the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers and the ADCSR.DBLE bit and TSSA bit should be set to all 0 in single scan mode.
The OCSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the internal reference voltage, the ADDISCR.ADNDIS[4:0] bits should be automatically set to 0Fh to discharge the A/D converter before sampling. The sampling time should be 5 µs or longer.
Sampling starts after discharging is completed during A/D conversion of the internal reference voltage, so an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

## 44.2.14 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, L, T, O)

Address(es): S12AD.ADSSTRL 0008 90DDh, S12AD.ADSSTRT 0008 90DEh, S12AD.ADSSTRO 0008 90DFh, S12AD.ADSSTR0 0008 90E0h, S12AD.ADSSTR1 0008 90E1h, S12AD.ADSSTR2 0008 90E2h, S12AD.ADSSTR3 0008 90E3h, S12AD.ADSSTR4 0008 90E4h, S12AD.ADSSTR5 0008 90E5h, S12AD.ADSSTR6 0008 90E6h, S12AD.ADSSTR7 0008 90E7h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

The ADSSTRn register sets the sampling time for analog input.
If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 54 MHz, one state is 18.5 ns. The initial value is 13 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0. The lower-limit value for sampling time differs depending on the PCLK to ADCLK frequency ratio.
Set a value that is 5 states or more when PCLK to ADCLK frequency ratio = 1:1, 2:1, 4:1, or 8:1.
Set a value that is 6 states or more when PCLK to ADCLK frequency ratio = 1:2 or 1:4.

Table 44.8 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 44.3.6, Analog Input Sampling Time and Scan Conversion Time.

**Table 44.8 Relationship between A/D Sampling State Register and Relevant Channels**

| Register Name | Channels |
|---|---|
| ADSSTR0 | AN000 |
| ADSSTR1 | AN001 |
| ADSSTR2 | AN002 |
| ADSSTR3 | AN003 |
| ADSSTR4 | AN004 |
| ADSSTR5 | AN005 |
| ADSSTR6 | AN006 |
| ADSSTR7 | AN007 |
| ADSSTRL | AN016 to AN020, AN027 |
| ADSSTRT | Temperature sensor output*1 |
| ADSSTRO | Internal reference voltage*1 |

Note 1. When performing A/D conversion of the temperature sensor output or internal reference voltage, the sampling time should be 5 µs or longer. Since the maximum number of states that can be set by this register is 255, take note of the ADCLK frequency. For example, when ADCLK = 54 MHz, the sampling time does not reach 5 µs even if 255 states is set.

## 44.2.15 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): S12AD.ADDISCR 0008 907Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | ADNDIS[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | ADNDIS[4:0] | A/D Disconnection Detection Assist Setting | b4 ADNDIS[4]: Discharge/precharge selected<br>0: Discharge<br>1: Precharge<br>b3 to b0 ADNDIS[3:0]: Discharge/precharge period | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADDISCR sets the disconnection detection assist function.

**ADNDIS[4:0] Bits (A/D Disconnection Detection Assist Setting)**

These bits select either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the ADNDIS[4] bit = 1 allows to select precharge and setting the ADNDIS[4] bit = 0 allows to select discharge. The period of precharge/discharge can be set with the ADNDIS[3:0] bits. When the ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is not effective. Setting of the ADNDIS[3:0] bits to 0001b is prohibited. Except for the case of ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge/discharge.

When the ADEXICR.OCSA or TSSA bit is set to 1 to perform A/D conversion of the temperature sensor output or internal reference voltage, ADNDIS[4:0] are automatically fixed to 0Fh, and discharging is executed prior to A/D conversion (auto-discharging). An auto-discharge period of 15 ADCLK cycles is inserted before sampling each time the temperature sensor output or internal reference voltage is A/D-converted.

## 44.2.16 A/D Event Link Control Register (ADELCCR)

Address(es): S12AD.ADELCCR 0008 907Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | ELCC[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | ELCC[1:0] | Event Link Control | b1 b0<br>0 0: Event is generated on completion of the scan other than group B in group scan mode<br>0 1: Event is generated on completion of the scan of group B in group scan mode<br>1 x: Event is generated on completion of all scans | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

x: Don't care

ADELCCR sets the generation conditions of the ELC scan end event (S12ADELC).

**ELCC[1:0] Bits (Event Link Control)**

These bits select the generation conditions of the scan end event (S12ADELC) for the ELC.

## 44.2.17 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): S12AD.ADGSPCR 0008 9080h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GBRP | — | — | — | — | — | — | — | — | — | — | — | — | — | GBRSCN | PGS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PGS | Group-A Priority Control Setting *1 | 0: Operation is without group-A priority control<br>1: Operation is with group-A priority control | R/W |
| b1 | GBRSCN | Group B Restart Setting *2 | (Enabled only when PGS = 1. Reserved when PGS = 0.)<br>0: Scanning for group B is not restarted after having been discontinued due to group-A priority control.<br>1: Scanning for group B is restarted after having been discontinued due to group-A priority control. | R/W |
| b14 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 | GBRP | Group B Single Scan Continuous Start *3 | (Enabled only when PGS = 1. Reserved when PGS = 0.)<br>0: Single scan for group B is not continuously activated.<br>1: Single scan for group B is continuously activated. | R/W |

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.
Note 2. When the GBRSCN bit is to be set to 1, the frequency ratio of peripheral module clock PCLK to A/D conversion clock ADCLK should be set to 1:1.
Note 3. When the GBRP bit has been set to 1, single scan is performed continuously for group B regardless of the setting of the GBRSCN bit.

ADGSPCR is used to make settings for priority control of A/D conversion for group A in group scan mode.

### PGS Bit (Group-A Priority Control Setting)

This bit sets the priority of operation on group A. Set this bit to 1 when giving priority to operation on group A.
When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode).
When setting the PGS bit to 0, clearing should be performed by software according to section 44.8.2, Notes on Stopping A/D Conversion. When setting the PGS bit to 1, follow the procedure described in section 44.3.4.3, Operation under Group-A Priority Control.

### GBRSCN Bit (Group B Restart Setting)

This bit controls the restarting of scan operation on group B when operation on group A is given priority.
If a scan operation on group B has been stopped by a group A trigger input with the GBRSCN bit set to 1, the scan operation is restarted on completion of the A/D conversion on group A. Also, if a group B trigger is input during A/D conversion on group A, the scan operation on group B is restarted on completion of the A/D conversion on group A.
If the GBRSCN bit has been set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.
The setting of the GBRSCN bit is enabled when the PGS bit is set to 1.

### GBRP Bit (Group B Single Scan Continuous Start)

This bit is set when a single scan operation is to be performed continuously on group B.
Setting the GBRP bit to 1 starts a single scan on group B. On completion of the scan, another single scan on group B is automatically started. If an A/D conversion on group B has been stopped due to an operation on group A that takes priority, single scan on group B is automatically restarted on completion of the A/D conversion on group A.
Disable group B trigger input before setting the GBRP bit to 1. Setting the GBRP bit to 1 invalidates the setting of the

GBRSCN bit. The ADCSR.ADST bit must be 0 when the GBRP bit is to be set.
The setting of the GBRP bit is enabled when the PGS bit is 1.

## 44.2.18 A/D Compare Function Control Register (ADCMPCR)

Address(es): S12AD.ADCMPCR 0008 9090h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | WCMPE | — | — | CMPAE | — | CMPBE | — | — | — | — | — | — | — | CMPAB[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CMPAB[1:0] | Window A/B Composite Condition Setting | b1 b0<br>0 0: S12ADWMELC is output when window A comparison conditions are met OR window B comparison conditions are met. S12ADWUMELC is output in other cases.<br>0 1: S12ADWMELC is output when window A comparison conditions are met EXOR window B comparison conditions are met. S12ADWUMELC is output in other cases.<br>1 0: S12ADWMELC is output when window A comparison conditions are met AND window B comparison conditions are met. S12ADWUMELC is output in other cases.<br>1 1: Setting prohibited. | R/W |
| b8 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b9 | CMPBE | Compare Window B Operation Enable | 0: Compare window B operation is disabled. S12ADWMELC and S12ADWUMELC outputs are disabled.<br>1: Compare window B operation is enabled. | R/W |
| b10 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b11 | CMPAE | Compare Window A Operation Enable | 0: Compare window A operation is disabled. S12ADWMELC and S12ADWUMELC outputs are disabled.<br>1: Compare window A operation is enabled. | R/W |
| b13, 12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b14 | WCMPE | Window Function Setting | 0: Window function is disabled. Window A and window B operate as a comparator to compare the single value on the lower side with the A/D conversion result.<br>1: Window function is enabled. Window A and window B operate as a comparator to compare the two values on the upper and lower sides with the A/D conversion result. | R/W |
| b15 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

ADCMPCR sets the compare window A and window B functions.

### CMPAB[1:0] Bits (Window A/B Composite Condition Setting)

These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits are used to select compare function match/mismatch event output conditions for the ELC or monitoring conditions of ADWINMON.MONCOMB. The CMPAB[1:0] bits should be set while the ADCSR.ADST bit is 0.

### CMPBE Bit (Compare Window B Operation Enable)

This bit enables or disables the compare window B operation. The CMPBE bit should be set while the ADCSR.ADST bit is 0.
Set this bit to 0 before setting the following registers.

- A/D channel select registers A0/A1/B0/B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- OCSA or TSSA in the A/D conversion extended input control register (ADEXICR.OCSA, TSSA)
- CMPCHB[5:0] in the window B channel select register (ADCMPBNSR.CMPCHB[5:0])

#### CMPAE Bit (Compare Window A Operation Enable)

This bit enables or disables the compare window A operation. The CMPAE bit should be set while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers.

- A/D channel select registers A0/A1/B0/B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- OCSA or TSSA in the A/D conversion extended input control register (ADEXICR.OCSA, TSSA)
- Window A channel select registers 0/1 (ADCMPANSR0, ADCMPANSR1)
- Window A extended input select register (ADCMPANSER)

#### WCMPE Bit (Window Function Setting)

This bit enables or disables the window function. The WCMPE bit should be set while the ADCSR.ADST bit is 0.

## 44.2.19 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0)

Address(es): S12AD.ADCMPANSR0 0008 9094h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CMPCHA007 | CMPCHA006 | CMPCHA005 | CMPCHA004 | CMPCHA003 | CMPCHA002 | CMPCHA001 | CMPCHA000 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPCHA000 | Compare Window A Channel Select | 0: The corresponding channel from among AN000 to AN007 is not a target for compare window A.<br>1: The corresponding channel from among AN000 to AN007 is a target for compare window A. | R/W |
| b1 | CMPCHA001 | | | R/W |
| b2 | CMPCHA002 | | | R/W |
| b3 | CMPCHA003 | | | R/W |
| b4 | CMPCHA004 | | | R/W |
| b5 | CMPCHA005 | | | R/W |
| b6 | CMPCHA006 | | | R/W |
| b7 | CMPCHA007 | | | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADCMPANSR0 is used to select analog input channels for comparison under compare window A conditions from among AN000 to AN007.

**CMPCHA0n Bit (n = 00 to 07) (Compare Window A Channel Select)**

Setting the CMPCHA0n bit which has the same number as the A/D channel selected by the ADANSA0.ANSA0n or ADANSB0.ANSB0n bit (n = 00 to 07) to 1 enables the compare function.

The CMPCHA0n bit should be set while ADCSR.ADST bit is 0.

## 44.2.20 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1)

Address(es): S12AD.ADCMPANSR1 0008 9096h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CMPCHA111 | — | — | — | — | — | — | CMPCHA104 | CMPCHA103 | CMPCHA102 | CMPCHA101 | CMPCHA100 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPCHA100 | Compare Window A Channel Select | 0: The corresponding channel from among AN016 to AN020 is not a target for compare window A.<br>1: The corresponding channel from among AN016 to AN020 is a target for compare window A. | R/W |
| b1 | CMPCHA101 | | | R/W |
| b2 | CMPCHA102 | | | R/W |
| b3 | CMPCHA103 | | | R/W |
| b4 | CMPCHA104 | | | R/W |
| b10 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | CMPCHA111 | Compare Window A Channel Select | 0: The corresponding channel from among AN027 is not a target for compare window A.<br>1: The corresponding channel from among AN027 is a target for compare window A. | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADCMPANSR1 is used to select analog input channels for comparison under compare window A conditions from among AN016 to AN020 and AN027.

**CMPCHA1n Bit (n = 00 to 04, 11) (Compare Window A Channel Select)**

Setting the CMPCHA1n which has the same number as the A/D channel selected by the ADANSA1.ANSA1n or ADANSB1.ANSB1n bit (n = 00 to 04, 11) to 1 enables the compare function.
The CMPCHA1n bit should be set while ADCSR.ADST bit is 0.

## 44.2.21 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER)

Address(es): S12AD.ADCMPANSER 0008 9092h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CMPOCA | CMPTSA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPTSA | Temperature Sensor Output Compare Select | 0: Temperature sensor output is not a target for compare window A.<br>1: Temperature sensor output is a target for compare window A. | R/W |
| b1 | CMPOCA | Internal Reference Voltage Compare Select | 0: Internal reference voltage is not a target for compare window A.<br>1: Internal reference voltage is a target for compare window A. | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

ADCMPANSER is used to select whether the temperature sensor output or internal reference voltage is compared under compare window A conditions.

**CMPTSA Bit (Temperature Sensor Output Compare Select)**

Setting the CMPTSA bit to 1 while the ADEXICR.TSSA bit is 1 enables the compare window A function. This bit should be set while the ADCSR.ADST bit is 0.

**CMPOCA Bit (Internal Reference Voltage Compare Select)**

Setting the CMPOCA bit to 1 while ADEXICR.OCSA bit is 1 enables the compare window A function. This bit should be set while the ADCSR.ADST bit is 0.

## 44.2.22 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0)

Address(es): S12AD.ADCMPLR0 0008 9098h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CMPLCHA007 | CMPLCHA006 | CMPLCHA005 | CMPLCHA004 | CMPLCHA003 | CMPLCHA002 | CMPLCHA001 | CMPLCHA000 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPLCHA000 | Compare Window A Comparison Condition Select | When the window function is disabled (ADCMPCR.WCMPE bit = 0):<br>0: ADCMPDR0 register value > A/D-converted value<br>1: ADCMPDR0 register value < A/D-converted value<br>When the window function is enabled (ADCMPCR.WCMPE bit = 1):<br>0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value<br>1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value | R/W |
| b1 | CMPLCHA001 | | | R/W |
| b2 | CMPLCHA002 | | | R/W |
| b3 | CMPLCHA003 | | | R/W |
| b4 | CMPLCHA004 | | | R/W |
| b5 | CMPLCHA005 | | | R/W |
| b6 | CMPLCHA006 | | | R/W |
| b7 | CMPLCHA007 | | | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The ADCMPLR0 register sets the condition for use in comparing the values of the ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion. The ADCMPLR0 register should be set while ADCSR.ADST bit is 0.

**CMPLCHA0n Bit (n = 00 to 07) (Compare Window A Comparison Condition Select)**

This bit sets the condition for use in comparison with the selected channel from among AN000 to AN007 to which compare window A conditions are applied. A condition can be set for individual comparison of each analog input. The CMPLCHA000 bit is used for AN000, the CMPLCHA007 bit is used for AN007.
When the result of comparison matches the set condition, the ADCMPSR0.CMPSTCHA0n flag is set to 1.
Figure 44.3 shows the comparison conditions.

(1) Comparison conditions when the window function is disabled

CMPLCHA0n = 0

| | |
|---|---|
| ADCMPDR0 value ≤ A/D converted value | Not met |
| ADCMPDR0 value > A/D converted value | Met |

CMPLCHA0n = 1

| | |
|---|---|
| ADCMPDR0 value < A/D converted value | Met |
| ADCMPDR0 value ≥ A/D converted value | Not met |

(2) Comparison conditions when the window function is enabled

CMPLCHA0n = 0

| | |
|---|---|
| ADCMPDR1 value < A/D-converted value | Met |
| ADCMPDR0 value ≤ A/D-converted value ≤ ADCMPDR1 value | Not met |
| A/D-converted value < ADCMPDR0 value | Met |

CMPLCHA0n = 1

| | |
|---|---|
| ADCMPDR1 value ≤ A/D-converted value | Not met |
| ADCMPDR0 value < A/D-converted value < ADCMPDR1 value | Met |
| A/D-converted value ≤ ADCMPDR0 value | Not met |

**Figure 44.3 Explanation of Compare Function Window A Comparison Conditions**

## 44.2.23 A/D Compare Function Window A Comparison Condition Setting Register 1 (ADCMPLR1)

Address(es): S12AD.ADCMPLR1 0008 909Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CMPLCHA111 | — | — | — | — | — | — | CMPLCHA104 | CMPLCHA103 | CMPLCHA102 | CMPLCHA101 | CMPLCHA100 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPLCHA100 | Compare Window A Comparison Condition Select | When the window function is disabled (ADCMPCR.WCMPE bit = 0):<br>0: ADCMPDR0 register value > A/D-converted value<br>1: ADCMPDR0 register value < A/D-converted value<br>When the window function is enabled (ADCMPCR.WCMPE bit = 1):<br>0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value<br>1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value | R/W |
| b1 | CMPLCHA101 | | | R/W |
| b2 | CMPLCHA102 | | | R/W |
| b3 | CMPLCHA103 | | | R/W |
| b4 | CMPLCHA104 | | | R/W |
| b10 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | CMPLCHA111 | Compare Window A Comparison Condition Select | When the window function is disabled (ADCMPCR.WCMPE bit = 0):<br>0: ADCMPDR0 register value > A/D-converted value<br>1: ADCMPDR0 register value < A/D-converted value<br>When the window function is enabled (ADCMPCR.WCMPE bit = 1):<br>0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value<br>1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The ADCMPLR1 register sets the condition for use in comparing the values of the ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.
The ADCMPLR1 register should be set while ADCSR.ADST bit is 0.

**CMPLCHA1n Bit (n = 00 to 04, 11) (Compare Window A Comparison Condition Select)**

This bit sets the condition for use in comparison with the selected channel from among AN016 to AN020 and AN027 to which compare window A conditions are applied. A condition can be set for individual comparison of each analog input. The CMPLCHA100 bit is used for AN016 and the CMPLCHA111 bit is used for AN027.
When the result of comparison matches the set condition, the ADCMPSR1.CMPSTCHA1n flag is set to 1.
Figure 44.3 shows the comparison conditions.

## 44.2.24 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER)

Address(es): S12AD.ADCMPLER 0008 9093h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CMPLOCA | CMPLTSA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPLTSA | Compare Window A Temperature Sensor Output Comparison Condition Select | When the window function is disabled (ADCMPCR.WCMPE bit = 0):<br>0: ADCMPDR0 register value > A/D-converted value<br>1: ADCMPDR0 register value < A/D-converted value<br><br>When the window function is enabled (ADCMPCR.WCMPE bit = 1):<br>0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value<br>1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value | R/W |
| b1 | CMPLOCA | Internal Reference Voltage Comparison Condition Select | When the window function is disabled (ADCMPCR.WCMPE bit = 0):<br>0: ADCMPDR0 register value > A/D-converted value<br>1: ADCMPDR0 register value < A/D-converted value<br><br>When the window function is enabled (ADCMPCR.WCMPE bit = 1):<br>0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value<br>1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The ADCMPLER register sets the condition for use in comparing the values of ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.
The ADCMPLER register should be set while ADCSR.ADST is 0.

### CMPLTSA Bit (Compare Window A Temperature Sensor Output Comparison Condition Select)

This bit sets the condition for use in comparison with temperature sensor output to which compare window A conditions are applied.
When the result of comparison matches the set condition, the ADCMPSER.CMPSTTSA flag is set to 1.
Figure 44.3 shows the comparison conditions.

### CMPLOCA Bit (Internal Reference Voltage Comparison Condition Select)

This bit sets conditions for use in comparison with internal reference voltage to which compare window A conditions are applied.
When the result of comparison matches the set condition, the ADCMPSER.CMPSTOCA flag is set to 1.
Figure 44.3 shows the comparison conditions.

## 44.2.25 A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0)

Address(es): S12AD.ADCMPDR0 0008 909Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ADCMPDR0 is a readable/writable register that sets the reference data when the compare window A function is used. ADCMPDR0 sets the lower-side level of window A.
The ADCMPDR0 register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value).

The ADCMPDR0 register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)

Set bits 15 to 2 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.

- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADCMPDR0 register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADCMPDR0 register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

## 44.2.26 A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1)

Address(es): S12AD.ADCMPDR1 0008 909Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ADCMPDR1 is a readable/writable register that sets the reference data when the compare window A function is used. ADCMPDR1 sets the upper-side level of window A.
The ADCMPDR1 register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value).
The ADCMPDR1 register is not used when the window function is disabled.

The ADCMPDR1 register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
- Set bits 15 to 2 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADCMPDR1 register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADCMPDR1 register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

## 44.2.27 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0)

Address(es): S12AD.ADCMPSR0 0008 90A0h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CMPST CHA007 | CMPST CHA006 | CMPST CHA005 | CMPST CHA004 | CMPST CHA003 | CMPST CHA002 | CMPST CHA001 | CMPST CHA000 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPSTCHA000 | Compare Window A Flag | When window A operation is enabled (ADCMPCR.CMPAE = 1), these flags indicate the comparison result of channels (AN000 to AN007 to which window A comparison conditions are applied.<br>0: Comparison conditions are not met.<br>1: Comparison conditions are met. | R/W |
| b1 | CMPSTCHA001 | | | R/W |
| b2 | CMPSTCHA002 | | | R/W |
| b3 | CMPSTCHA003 | | | R/W |
| b4 | CMPSTCHA004 | | | R/W |
| b5 | CMPSTCHA005 | | | R/W |
| b6 | CMPSTCHA006 | | | R/W |
| b7 | CMPSTCHA007 | | | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The ADCMPSR0 register stores the comparison results of the compare window A function.

**CMPSTCHA0n Flag (n = 00 to 07) (Compare Window A Flag)**

This flag is comparison result status flag of channel (AN000 to AN007) to which window A comparison conditions are applied. When the comparison condition set by ADCMPLR0.CMPLCHAn is met at the end of A/D conversion, each of these flags is set to 1. CMPSTCHA000 and CMPSTCHA007 correspond to AN000 and AN007, respectively.
Writing 1 to the CMPSTCHA0n flag is disabled.
[Setting condition]

- The condition set by ADCMPLR0.CMPLCHA0n is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

## 44.2.28 A/D Compare Function Window A Channel Status Register 1 (ADCMPSR1)

Address(es): S12AD.ADCMPSR1 0008 90A2h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CMPST CHA111 | — | — | — | — | — | — | CMPST CHA104 | CMPST CHA103 | CMPST CHA102 | CMPST CHA101 | CMPST CHA100 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPSTCHA100 | Compare Window A Flag | When window A operation is enabled (ADCMPCR.CMPAE = 1), these flags indicate the comparison result of channels (AN016 to AN020) to which window A comparison conditions are applied.<br>0: Comparison conditions are not met.<br>1: Comparison conditions are met. | R/W |
| b1 | CMPSTCHA101 | | | R/W |
| b2 | CMPSTCHA102 | | | R/W |
| b3 | CMPSTCHA103 | | | R/W |
| b4 | CMPSTCHA104 | | | R/W |
| b10 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | CMPSTCHA111 | Compare Window A Flag | When window A operation is enabled (ADCMPCR.CMPAE = 1), this flag indicates the comparison result of channel (AN027) to which window A comparison condition is applied.<br>0: Comparison conditions are not met.<br>1: Comparison conditions are met. | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The ADCMPSR1 register stores the comparison results of the compare window A function.

**CMPSTCHA1n Flag (n = 00 to 04, 11) (Compare Window A Flag)**

This flag is comparison result status flag of channel (AN016 to AN020, AN027) to which window A comparison conditions are applied. When the comparison condition set by ADCMPLR1.CMPLCHA1n is met at the end of A/D conversion, each of these flags is set to 1. The CMPSTCHA100 bit corresponds to AN016, the CMPSTCHA104 bit corresponds to AN020, and the CMPSTCHA111 bit corresponds to AN027.
Writing 1 to the CMPSTCHA1n flag (n = 00 to 04, 11) is disabled.
[Setting condition]

- The condition set by ADCMPLR1.CMPLCHA1n is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

## 44.2.29 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): S12AD.ADCMPSER 0008 90A4h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CMPSTOCA | CMPSTTSA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPSTTSA | Compare Window A Temperature Sensor Output Compare Flag | When window A operation is enabled (ADCMPCR.CMPAE = 1), this flag indicates the temperature sensor output comparison result.<br>0: Comparison conditions are not met.<br>1: Comparison conditions are met. | R/W |
| b1 | CMPSTOCA | Compare Window A Internal Reference Voltage Compare Flag | When window A operation is enabled (ADCMPCR.CMPAE = 1), this flag indicates the internal reference voltage comparison result.<br>0: Comparison conditions are not met.<br>1: Comparison conditions are met. | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The ADCMPSER register stores the comparison result of the compare window A function.

**CMPSTTSA Flag (Compare Window A Temperature Sensor Output Compare Flag)**

This flag is a status flag that indicates the temperature sensor output comparison result. When the comparison condition set by ADCMPLER.CMPLTSA is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTTSA flag is disabled.
[Setting condition]

- The condition set by ADCMPLER.CMPLTSA is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

**CMPSTOCA Flag (Compare Window A Internal Reference Voltage Compare Flag)**

This flag is a status flag that indicates the internal reference voltage comparison result. When the comparison condition set by ADCMPLER.CMPLOCA is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTOCA flag is disabled.
The value 1 cannot be written to the CMPSTOCA bit.
[Setting condition]

- The condition set in by ADCMPLER.CMPLOCA bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

## 44.2.30 A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT)

Address(es): S12AD.ADHVREFCNT 0008 908Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ADSLP | — | — | LVSEL | — | — | HVSEL[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | HVSEL[1:0] | High-Potential Reference Voltage Select | b1 b0<br>0 0: AVCC0 is selected as the high-potential reference voltage.<br>0 1: VREFH0 is selected as the high-potential reference voltage.<br>Settings other than above are prohibited. | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | LVSEL | Low-Potential Reference Voltage Select | 0: AVSS0 is selected as the low-potential reference voltage.<br>1: VREFL0 is selected as the low-potential reference voltage. | R/W |
| b6, b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | ADSLP | Sleep | 0: Normal operation<br>1: Standby state | R/W |

The ADHVREFCNT register specifies the high-potential and low-potential reference voltages. Set this register before performing A/D conversion.

### HVSEL[1:0] Bits (High-Potential Reference Voltage Select)

These bits are used to set the high-potential reference voltage. AVCC0 or VREFH0 is selectable as the high-potential reference voltage.

### LVSEL Bit (Low-Potential Reference Voltage Select)

This bit is used to set the low-potential reference voltage. AVSS0 or VREFL0 is selectable as the low-potential reference voltage.

### ADSLP Bit (Sleep)

This bit is used to transition the 12-bit A/D converter to the standby state. Set the ADSLP bit to 1 only when modifying the ADCSR.ADHSC bit. In other cases, setting the ADSLP bit to 1 is prohibited.

After the ADSLP bit is set to 1, wait at least 5 µs before clearing this bit to 0. Furthermore, after the ADSLP bit is cleared to 0, wait at least 1 µs and then start the A/D conversion.

For the ADHSC bit rewriting procedure, see section 44.8.10, ADHSC Bit Rewriting Procedure.

## 44.2.31 A/D Compare Function Window A/B Status Monitor Register (ADWINMON)

Address(es): S12AD.ADWINMON 0008 908Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | MONCMPB | MONCMPA | — | — | — | MONCOMB |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | MONCOMB | Combination Result Monitor Flag | This flag indicates the combination result.<br>This flag is valid when both window A operation and window B operation are enabled.<br>0: Window A/window B composite conditions are not met.<br>1: Window A/window B composite conditions are met. | R |
| b3 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R |
| b4 | MONCMPA | Comparison Result Monitor A Flag | 0: Window A comparison conditions are not met.<br>1: Window A comparison conditions are met. | R |
| b5 | MONCMPB | Comparison Result Monitor B Flag | 0: Window B comparison conditions are not met.<br>1: Window B comparison conditions are met. | R |
| b7, b6 | — | Reserved | These bits are read as 0. | R |

The ADWINMON register can monitor the comparison result and the combination result.

### MONCOMB Flag (Combination Result Monitor Flag)

This read-only flag indicates the result in combination of comparison condition result A and comparison result condition B with the combination condition set by the ADCMPCR.CMPAB[1:0] bits.
[Setting condition]

- The combined result meets the combination condition set by the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set by the ADCMPCR.CMPAB[1:0] bits.
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

### MONCMPA Flag (Comparison Result Monitor A Flag)

This read-only flag is read as 1 when the A/D-converted value of the window A target channel meets the condition set in the ADCMPLR0, ADCMPLR1, and ADCMPLER registers, and is read as 0 otherwise.
[Setting condition]

- The A/D-converted value meets the condition set by the ADCMPLR0.CMPLCHA0n bit when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set by the ADCMPLR0.CMPLCHA0n bit when ADCMPCR.CMPAE = 1.
- ADCMPCR.CMPAE = 0 (Automatically cleared when the ADCMPCR.CMPAE bit value changes from 1 to 0.)

### MONCMPB Flag (Comparison Result Monitor B Flag)

This read-only flag is read as 1 when the A/D converted value of the window B target channel meets the condition set by the ADCMPBNSR.CMPLB bit, and is read as 0 in other cases.
[Setting condition]

- The A/D-converted value meets the condition set by ADCMPBNSR.CMPLB bit when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set by ADCMPBNSR.CMPLB bit when ADCMPCR.CMPBE = 1.
- ADCMPCR.CMPBE = 0 (Automatically cleared when the ADCMPCR.CMPBE bit value changes from 1 to 0.)

## 44.2.32 A/D Compare Function Window B Channel Select Register (ADCMPBNSR)

Address(es): S12AD.ADCMPBNSR 0008 90A6h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CMPLB | — | CMPCHB[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | CMPCHB[5:0] | Compare Window B Channel Select | These bits select channels to be compared with the compare window B conditions.<br>b5 b0<br>0 0 0 0 0 0: AN000<br>0 0 0 0 0 1: AN001<br>0 0 0 0 1 0: AN002<br>: :<br>0 0 0 1 1 0: AN006<br>0 0 0 1 1 1: AN007<br>0 1 0 0 0 0: AN016<br>0 1 0 0 0 1: AN017<br>: :<br>0 1 0 1 0 0: AN020<br>0 1 1 0 1 1: AN027<br>1 0 0 0 0 0: Temperature sensor<br>1 0 0 0 0 1: Internal reference voltage<br>Settings other than above are prohibited. | R/W |
| b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b7 | CMPLB | Compare Window B Comparison Condition Setting | When the window function is disabled (ADCMPCR.WCMPE bit = 0)<br>0: ADWINLLB register value > A/D-converted value<br>1: ADWINLLB register value < A/D-converted value<br><br>When the window function is enabled (ADCMPCR.WCMPE bit = 1)<br>0: A/D-converted value < ADWINLLB register value or ADWINULB register value < A/D-converted value<br>1: ADWINLLB register value < A/D-converted value < ADWINULB register value | R/W |

The ADCMPBNSR register is used to set the compare window B function.

### CMPCHB[5:0] Bits (Compare Window B Channel Select)

These bits are used to select channels to be compared from among AN000 to AN007, AN016 to AN020, AN027, temperature sensor, and internal reference voltage against the conditions of comparison window B.
The compare window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected by the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers.
The CMPCHB[5:0] bits should be set while the ADCSR.ADST bit is 0.

### CMPLB Bit (Compare Window B Comparison Condition Setting)

This bit is used to set comparison conditions of channels for window B. When the comparison result of each analog input meets the set condition, the ADCMPBSR.CMPSTB flag is set to 1.
Figure 44.4 shows the comparison conditions.

(1) Compare conditions when the window function is disabled

CMPLB = 0

| | |
|---|---|
| ADWINLLB value ≤ A/D-converted value | Not met |
| ADWINLLB value > A/D-converted value | Met |

CMPLB = 1

| | |
|---|---|
| ADWINLLB value < A/D-converted value | Met |
| ADWINLLB value ≥ A/D-converted value | Not met |

(2) Comparison conditions when the window function is enabled

CMPLB = 0

| | |
|---|---|
| A/D-converted value < ADWINULB value | Met |
| ADWINLLB value ≤ A/D-converted value ≤ ADWINULB value | Not met |
| A/D-converted value < ADWINLLB value | Met |

CMPLB = 1

| | |
|---|---|
| A/D-converted value ≤ ADWINULB value | Not met |
| ADWINLLB value < A/D-converted value < ADWINULB value | Met |
| A/D-converted value ≤ ADWINLLB value | Not met |

**Figure 44.4 Explanation of Compare Function Window B Compare Conditions**

## 44.2.33 A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB)

Address(es): S12AD.ADWINLLB 0008 90A8h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ADWINLLB is a readable/writable register that sets the reference data when the compare window B function is used. ADWINLLB sets the lower-side level of window B.
The ADWINLLB register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADWINULB setting value ≥ ADWINLLB setting value).

The ADWINLLB register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y (ADDRy) is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)

Set bits 15 to 2 to the lower-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.

- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the lower-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADWINLLB register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADWINLLB register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

## 44.2.34 A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB)

Address(es): S12AD.ADWINULB 0008 90AAh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ADWINULB is a readable/writable register that sets the reference data when the compare window B function is used. ADWINULB sets the upper-side level of window B.
The ADWINULB register is writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.
Set the registers so that the upper-side level is not less than the lower-side level (ADWINULB setting value ≥ ADWINLLB setting value)
The ADWINULB register is not used when the window function is disabled.

The ADWINULB register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

Note: If a format different from the format setting of A/D data register y (ADDRy) is used to set the compare value, a correct comparison result will not be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level. Write 0 to bits 3 to 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  Set bits 11 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 to 12.
- Flush-left format
  Set bits 15 to 4 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 3 to 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 13 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 15 and 14.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  Set bits 15 to 2 to the upper-side comparison level for comparison with the A/D-converted value of the same channel. Write 0 to bits 1 and 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  Set bits 15 to 0 to the upper-side comparison level for comparison with the A/D-converted value of the same channel.

When A/D-converted addition mode is selected, set the value added by the A/D-converted value of the same channel. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the A/D conversion count is set to 1 to 4 times, set the number of conversion accuracy bits extended by 2 bits in the ADWINULB register; when the A/D conversion count is set to 16 times, set the number of conversion accuracy bits extended by 4 bits in the ADWINULB register.
Even if A/D converted value addition mode is selected, set the reference data in the A/D data register according to the settings of the A/D data register format select bits.

## 44.2.35 A/D Compare Function Window B Channel Status Register (ADCMPBSR)

Address(es): S12AD.ADCMPBSR 0008 90ACh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CMPSTB |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPSTB | Compare Window B Flag | 0: Comparison conditions are not met.<br>1: Comparison conditions are met. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The ADCMPBSR register stores the comparison result of the compare window B function.

**CMPSTB Flag (Compare Window B Flag)**

This flag is a status flag that indicates the results of comparison for the channels (AN000 to AN007, AN016 to AN020, AN027, temperature sensor, and internal reference voltage) to which the window B comparison conditions are applied. When the comparison condition set by ADCMPBNSR.CMPCHB[5:0] bits is met at the end of A/D conversion, this flag is set to 1.
Writing 1 to the CMPSTB flag is disabled.
[Setting condition]

- The condition set by ADCMPBNSR.CMPLB bit is met when ADCMPCR.CMPAE = 1

[Clearing condition]

- 0 is written after reading 1

## 44.2.36 A/D Data Storage Buffer Register n (ADBUFn) (n = 0 to 15)

Address(es): S12AD.ADBUF0 0008 90B0h, S12AD.ADBUF1 0008 90B2h, S12AD.ADBUF2 0008 90B4h, S12AD.ADBUF3 0008 90B6h, S12AD.ADBUF4 0008 90B8h, S12AD.ADBUF5 0008 90BAh, S12AD.ADBUF6 0008 90BCh, S12AD.ADBUF7 0008 90BEh, S12AD.ADBUF8 0008 90C0h, S12AD.ADBUF9 0008 90C2h, S12AD.ADBUF10 0008 90C4h, S12AD.ADBUF11 0008 90C6h, S12AD.ADBUF12 0008 90C8h, S12AD.ADBUF13 0008 90CAh, S12AD.ADBUF14 0008 90CCh, S12AD.ADBUF15 0008 90CEh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A/D data storage buffer registers n (ADBUFn) are 16-bit read-only registers that sequentially store all A/D converted values. The automatic clear function is not applied to these registers.

The ADBUFn register uses different formats depending on the following conditions.

- Settings of the A/D data register format select bit (flush-right or flush-left)
- Settings of the A/D-converted value addition/average function select register (A/D-converted value average mode selected or not selected)
- Settings of the A/D-converted value addition/average count select register (addition/average mode selected, addition count selected)

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Value Average Mode is Selected

- Flush-right format
  The mean value of the A/D converted results of the same channel is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The mean value of the A/D converted results of the same channel is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

A/D-converted value average mode can be set only when two or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversion can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the ADBUFn