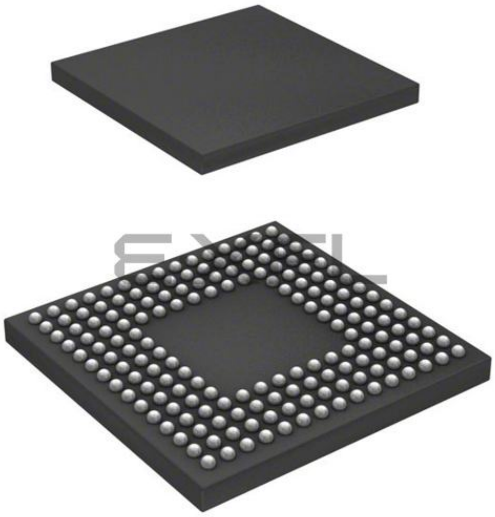


## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Not For New Designs |
| Core Processor | SH2A-FPU |
| Core Size | 32-Bit Single-Core |
| Speed | 200MHz |
| Connectivity | CANbus, Ethernet, I²C, SCI, SPI, USB |
| Peripherals | DMA, PWM, WDT |
| Number of I/O | 112 |
| Program Memory Size | 1MB (1M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 128K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 8x12b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 176-LFBGA |
| Supplier Device Package | 176-LFBGA (13x13) |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f72167adbg-u1 |

### 5.1.3 Exception Handling Vector Table

Before exception handling begins running, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception service routines. (The reset exception handling table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception handling, the start addresses of the exception service routines are fetched from the exception handling vector table, which is indicated by this vector table address.

Table 5.3 shows the vector numbers and vector table address offsets. Table 5.4 shows how vector table addresses are calculated.

**Table 5.3 Exception Handling Vector Table**

| Exception Sources | | Vector Numbers | Vector Table Address Offset |
|---|---|---|---|
| Power-on reset | PC | 0 | H'00000000 to H'00000003 |
| | SP | 1 | H'00000004 to H'00000007 |
| Manual reset | PC | 2 | H'00000008 to H'0000000B |
| | SP | 3 | H'0000000C to H'0000000F |
| General illegal instruction | | 4 | H'00000010 to H'00000013 |
| (Reserved by system) | | 5 | H'00000014 to H'00000017 |
| Slot illegal instruction | | 6 | H'00000018 to H'0000001B |
| (Reserved by system) | | 7 | H'0000001C to H'0000001F |
| | | 8 | H'00000020 to H'00000023 |
| CPU address error | | 9 | H'00000024 to H'00000027 |
| DMAC address error | | 10 | H'00000028 to H'0000002B |
| Interrupts | NMI | 11 | H'0000002C to H'0000002F |
| | User break | 12 | H'00000030 to H'00000033 |
| FPU exception | | 13 | H'00000034 to H'00000037 |
| H-UDI | | 14 | H'00000038 to H'0000003B |
| Bank overflow | | 15 | H'0000003C to H'0000003F |
| Bank underflow | | 16 | H'00000040 to H'00000043 |

### 6.8.1 Banked Register and Input/Output of Banks

#### (1) Banked Register

The contents of the general registers (R0 to R14), global base register (GBR), multiply and accumulate registers (MACH and MACL), and procedure register (PR), and the vector table address offset are banked.

#### (2) Register Banks

This LSI has fifteen register banks, bank 0 to bank 14. Register banks are stacked in first-in last-out (FILO) sequence. Saving takes place in order, beginning from bank 0, and restoration takes place in the reverse order, beginning from the last bank saved to.

### 6.8.2 Bank Save and Restore Operations

#### (1) Saving to Bank

Figure 6.11 shows register bank save operations. The following operations are performed when an interrupt for which usage of register banks is allowed is accepted by the CPU:

a. Assume that the bank number bit value in the bank number register (IBNR), BN, is "i" before the interrupt is generated.
b. The contents of registers R0 to R14, GBR, MACH, MACL, and PR, and the interrupt vector table address offset (VTO) of the accepted interrupt are saved in the bank indicated by BN, bank i.
c. The BN value is incremented by 1.

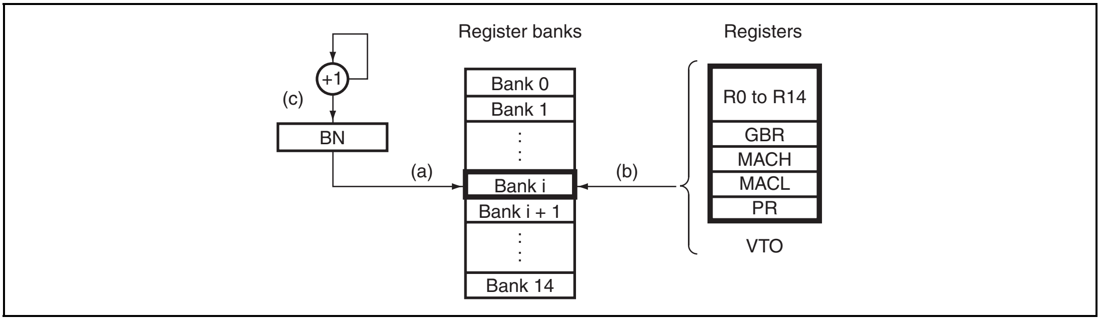


**Figure 6.11 Bank Save Operations**

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 21 to 19 | IWRRD[2:0] | 011 | R/W | Idle Cycles for Read-Read in Another Space<br>Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles switch between different spaces.<br>000: No idle cycle inserted<br>001: 1 idle cycle inserted<br>010: 2 idle cycles inserted<br>011: 4 idle cycles inserted<br>100: 6 idle cycles inserted<br>101: 8 idle cycles inserted<br>110: 10 idle cycles inserted<br>111: 12 idle cycles inserted |
| 18 to 16 | IWRRS[2:0] | 011 | R/W | Idle Cycles for Read-Read in the Same Space<br>Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles are for the same space.<br>000: No idle cycle inserted<br>001: 1 idle cycle inserted<br>010: 2 idle cycles inserted<br>011: 4 idle cycles inserted<br>100: 6 idle cycles inserted<br>101: 8 idle cycles inserted<br>110: 10 idle cycles inserted<br>111: 12 idle cycles inserted |
| 15 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

### Table 9.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-1

| Setting | | | | |
|---|---|---|---|---|
| BSZ [1:0] | A2/3 ROW [1:0] | A2/3 COL [1:0] | | |
| 11 (32 Bits) | 00 (11 Bits) | 00 (8 Bits) | | |
| Output Pin of This LSI | Row Address Output Cycle | Column Address Output Cycle | SDRAM Pin | Function |
| A17 | A25 | A17 | | Unused |
| A16 | A24 | A16 | | |
| A15 | A23 | A15 | | |
| A14 | A22*[2] | A22*[2] | A12(BA1) | Specifies bank |
| A13 | A21*[2] | A21*[2] | A11(BA0) | |
| A12 | A20 | L/H*[1] | A10/AP | Specifies address/precharge |
| A11 | A19 | A11 | A9 | Address |
| A10 | A18 | A10 | A8 | |
| A9 | A17 | A9 | A7 | |
| A8 | A16 | A8 | A6 | |
| A7 | A15 | A7 | A5 | |
| A6 | A14 | A6 | A4 | |
| A5 | A13 | A5 | A3 | |
| A4 | A12 | A4 | A2 | |
| A3 | A11 | A3 | A1 | |
| A2 | A10 | A2 | A0 | |
| A1 | A9 | A1 | | Unused |
| A0 | A8 | A0 | | |
| Example of connected memory | | | | |
| 64-Mbit product (512 Kwords × 32 bits × 4 banks, column 8 bits product): 1<br>16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 2 | | | | |

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

### Table 9.15 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-1

| Setting | | | | |
|---|---|---|---|---|
| BSZ [1:0] | A2/3 ROW [1:0] | A2/3 COL [1:0] | | |
| 10 (16 Bits) | 01 (12 Bits) | 01 (9 Bits) | | |
| Output Pin of This LSI | Row Address Output Cycle | Column Address Output Cycle | SDRAM Pin | Function |
| A17 | A26 | A17 | | Unused |
| A16 | A25 | A16 | | |
| A15 | A24 | A15 | | |
| A14 | A23*[2] | A23*[2] | A13 (BA1) | Specifies bank |
| A13 | A22*[2] | A22*[2] | A12 (BA0) | |
| A12 | A21 | A12 | A11 | Address |
| A11 | A20 | L/H*[1] | A10/AP | Specifies address/precharge |
| A10 | A19 | A10 | A9 | Address |
| A9 | A18 | A9 | A8 | |
| A8 | A17 | A8 | A7 | |
| A7 | A16 | A7 | A6 | |
| A6 | A15 | A6 | A5 | |
| A5 | A14 | A5 | A4 | |
| A4 | A13 | A4 | A3 | |
| A3 | A12 | A3 | A2 | |
| A2 | A11 | A2 | A1 | |
| A1 | A10 | A1 | A0 | |
| A0 | A9 | A0 | | Unused |
| Example of connected memory | | | | |
| 128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 1 | | | | |

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

### (3) Input Capture Function

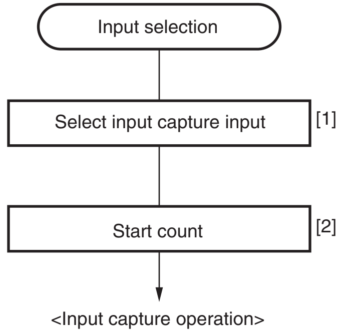
The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0 and 1, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 1, Pϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if Pϕ/1 is selected.

#### (a) Example of Input Capture Operation Setting Procedure

Figure 11.10 shows an example of the input capture operation setting procedure.



[1] Designate TGR as an input capture register by means of TIOR, and select rising edge, falling edge, or both edges as the input capture source and input signal edge.

[2] Set the CST bit in TSTR to 1 to start the count operation.

**Figure 11.10 Example of Input Capture Operation Setting Procedure**

### (21) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 11.159 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

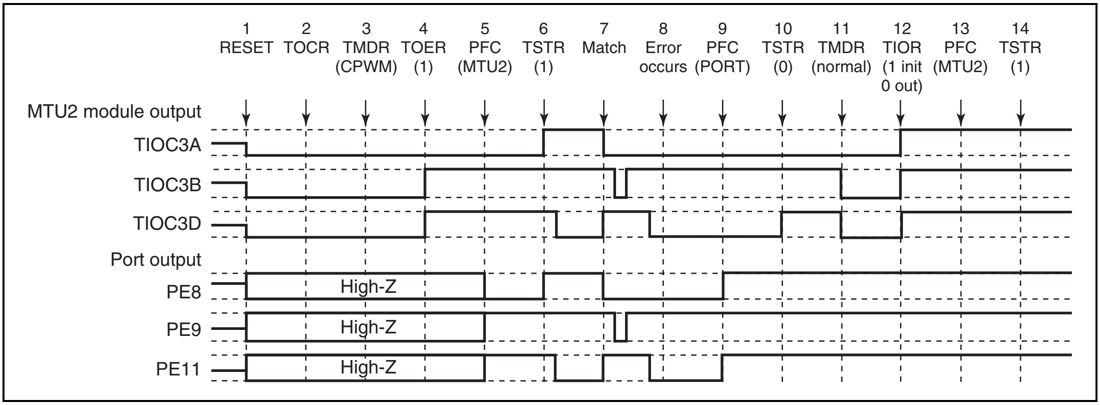


**Figure 11.159 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode**

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the complementary PWM output initial value.)
11. Set normal mode. (MTU2 output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

### 16.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

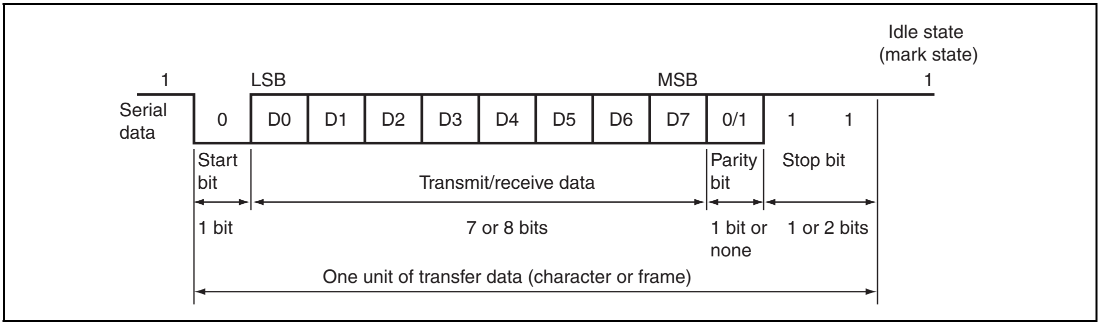


**Figure 16.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)**

# Section 17 Serial Communication Interface with FIFO (SCIF)

This LSI has one channel of serial communication interface with FIFO (SCIF) that supports both asynchronous and clocked synchronous serial communication. It also has 16-stage FIFO registers for both transmission and reception independently for each channel that enable this LSI to perform efficient high-speed continuous communication.

## 17.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communications interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level). It is also detected by reading the RXD level directly from the serial port register when a framing error occurs.
- Clocked synchronous serial communication:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clocked synchronous communication function. There is one serial data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)

### (2-5) Transfer Operation Flowchart

Figure 18.29 shows an example of transfer operation flowchart in master mode during clock synchronous operation.

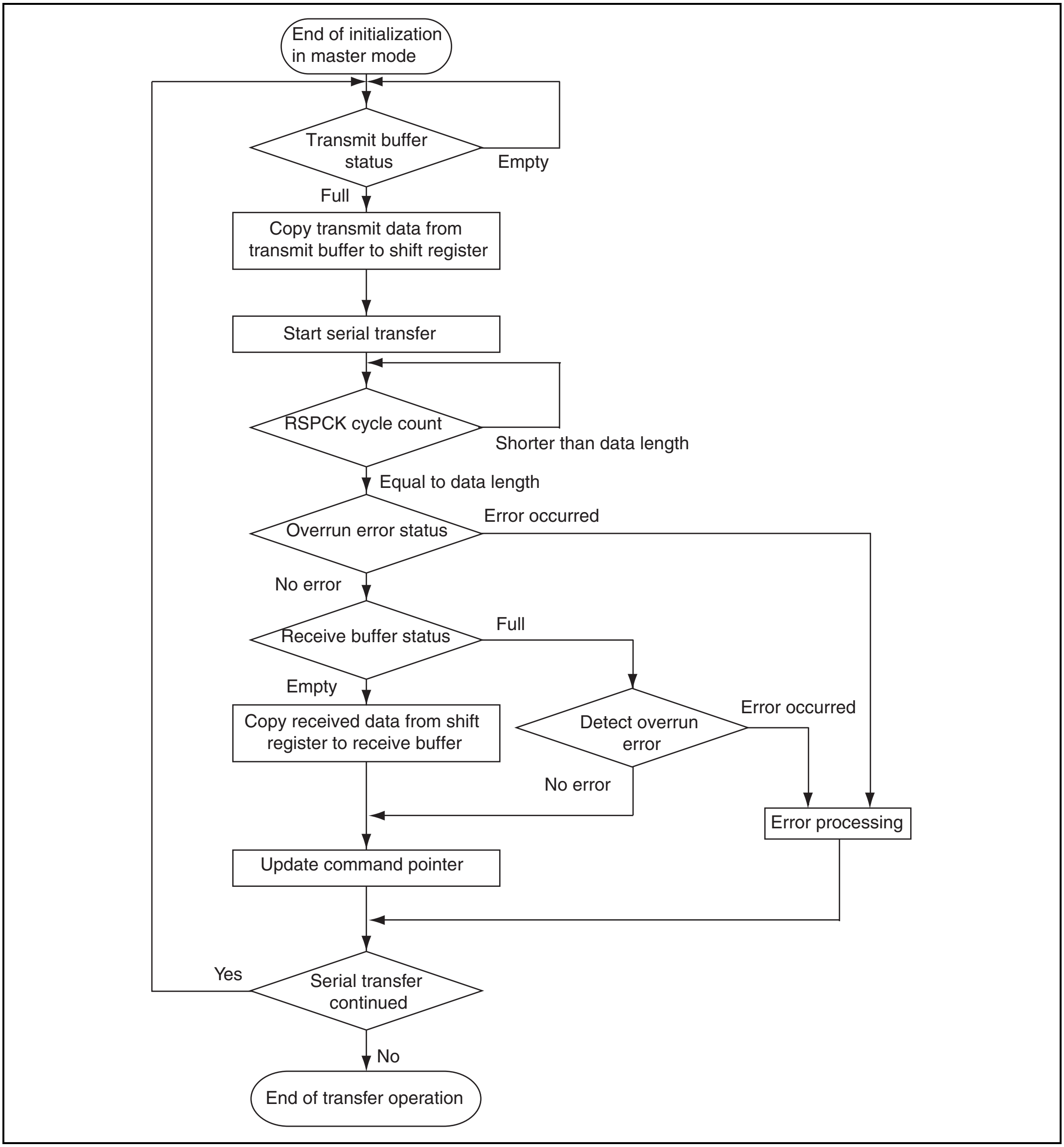


**Figure 18.29 Example of Transfer Operation Flowchart in Master Mode**

TSEG2: TSG2 + 1

BRP: BRP[7:0] (bits 7 to 0 in BCR0)

The RCAN-ET Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 * (BRP + 1) * (TSEG1 + TSEG2 + 1)}$$

where BRP is given by the register value, and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values. The '+ 1' in the above formula is for the Sync-Seg which duration is 1 time quanta.

$f_{CLK}$ = Peripheral Clock

BCR Setting Constraints

$TSEG1min > TSEG2 \geq SJWmax$ (SJW = 1 to 4)

$8 \leq TSEG1 + TSEG2 + 1 \leq 25$ time quanta ($TSEG1 + TSEG2 + 1 = 7$ is not allowed)

$TSEG2 \geq 2$

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

| Port | Function 1 (Related Module) | Function 2 (Related Module) | Function 3 (Related Module) | Function 4 (Related Module) | Function 5 (Related Module) | Function 6 (Related Module) | Function 7 (Related Module) | Function 8 (Related Module) |
|---|---|---|---|---|---|---|---|---|
| A | PA9 I/O (Port) | $\overline{\text{CS3}}$ output (BSC) | — | IRQ3 input (INTC) | TCLKD input (MTU2) | SSLO I/O (RSPI) | SCK0 I/O (SCI) | MII_TXD1 output (Ether) |
| | PA8 I/O (Port) | $\overline{\text{CS4}}$ output (BSC) | — | IRQ4 input (INTC) | TCLKC input (MTU2) | MISO I/O (RSPI) | RXD1 input (SCI) | MII_TXD2 output (Ether) |
| | PA7 I/O (Port) | $\overline{\text{CS5}}$ output (BSC) | — | IRQ5 input (INTC) | TCLKB input (MTU2) | MOSI I/O (RSPI) | TXD1 output (SCI) | MII_TXD3 output (Ether) |
| | PA6 I/O (Port) | $\overline{\text{CS6}}$ output (BSC) | — | IRQ6 input (INTC) | TCLKA input (MTU2) | RSPCK I/O (RSPI) | SCK1 I/O (SCI) | TX_ER output (Ether) |
| | PA5 I/O (Port) | $\overline{\text{CS5}}$ output (BSC) | — | — | TCLKA input (MTU2) | RSPCK I/O (RSPI) | SCK1 I/O (SCI) | RX_ER input (Ether) |
| | PA4 I/O (Port) | $\overline{\text{CS4}}$ output (BSC) | — | — | TCLKB input (MTU2) | MOSI I/O (RSPI) | TXD1 output (SCI) | MII_RXD3 input (Ether) |
| | PA3 I/O (Port) | $\overline{\text{CS3}}$ output (BSC) | — | — | TCLKC input (MTU2) | MISO I/O (RSPI) | RXD1 input (SCI) | MII_RXD2 input (Ether) |
| | PA2 I/O (Port) | $\overline{\text{CS2}}$ output (BSC) | — | — | TCLKD input (MTU2) | SSLO I/O (RSPI) | SCK0 I/O (SCI) | MII_RXD1 input (Ether) |
| | PA1 I/O (Port) | $\overline{\text{CS1}}$ output (BSC) | — | IRQ5 input (INTC) | — | CTx0 output (RCAN-ET) | TXD0 output (SCI) | MII_RXD0 input (Ether) |
| | PA0 I/O (Port) | $\overline{\text{CS0}}$ output (BSC) | — | IRQ4 input (INTC) | — | CRx0 input (RCAN-ET) | RXD0 input (SCI) | RX_CLK input (Ether) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 11 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 10 to 8 | PE10MD[2:0] | 000 | R/W | PE10 Mode<br>Select the function of the PE10/DREQ3/TIOC3C/SSL3/TXD2/TX_CLK pin.<br>000: PE10 I/O (port)<br>001: Setting prohibited<br>010: DREQ3 input (DMAC)<br>011: Setting prohibited<br>100: TIOC3C I/O (MTU2)<br>101: SSL3 output (RSPI)<br>110: TXD2 output (SCI)<br>111: TX_CLK input (Ether) |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 to 4 | PE9MD[2:0] | 000 | R/W | PE9 Mode<br>Select the function of the PE9/DACK2/TIOC3B/TX_EN pin.<br>000: PE9 I/O (port)<br>001: Setting prohibited<br>010: DACK2 output (DMAC)<br>011: Setting prohibited<br>100: TIOC3B I/O (MTU2)<br>101: Setting prohibited<br>110: Setting prohibited<br>111: TX_EN output (Ether) |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | PE5PR | Pin state | R | The pin state is returned regardless of the PFC setting. These bits cannot be modified. |
| 4 | PE4PR | Pin state | R | |
| 3 | PE3PR | Pin state | R | |
| 2 | PE2PR | Pin state | R | |
| 1 | PE1PR | Pin state | R | |
| 0 | PE0PR | Pin state | R | |

## 23.6 Port F

Port F is an I/O port with 8 pins shown in figure 23.6.

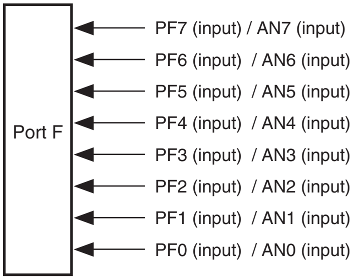

Port F
PF7 (input) / AN7 (input)
PF6 (input) / AN6 (input)
PF5 (input) / AN5 (input)
PF4 (input) / AN4 (input)
PF3 (input) / AN3 (input)
PF2 (input) / AN2 (input)
PF1 (input) / AN1 (input)
PF0 (input) / AN0 (input)

**Figure 23.6 Port F**

### 23.6.1 Register Descriptions

Port F has the following registers. See section 32, List of Registers for details on the register address and states in each operating mode.

**Table 23.11 Register Configuration**

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Port F data register L | PFDRL | R | — | H'FFFE3A82 | 8, 16 |

## 24.9 DTC Transfer

This module allows DTC transfer for endpoints 1, 2, 4, and 5, excluding transfer of word and longword. If endpoint 1 contains at least one byte of valid receive data, a DTC transfer request is issued to endpoint 1. If there is no valid data in endpoint 2, a DTC transfer request is issued to endpoint 2. If endpoint 4 contains at least one byte of valid receive data, a DTC transfer request is issued to endpoint 4. If there is no valid data in endpoint 5, a DTC transfer request is issued to endpoint 5.

When EP1DMAE or EP4DMAE in the USBDMA setting register is set to 1 to allow DTC transfer, 0-length data received for endpoint 1 or 4 is ignored. When DTC transfer is set, it is unnecessary to write 1 to the EP1RDFN, EP2PKTE, EP4RDFN, and EP5PKTE bits in USBTRG1 or USBTRG2. (However, the PKTE bit in USBTRG1 or USBTRG2 must be set to 1 for data with a size less than the maximum number of bytes.) For EP1 and EP4, the FIFO buffer automatically becomes empty when the received data has been completely read. For EP2 and EP5, the FIFO automatically becomes full when the maximum number of bytes (64 bytes) is written to the FIFO, allowing the data in the FIFO to be transmitted. (See figures 24.22 and 24.25.)

### 24.9.1 DTC Transfer for Endpoints 1 and 4

If the received data for EP1 is transferred by DTC, when the currently selected data FIFO becomes empty, processing equivalent to writing 1 to the EP1RDFN bit in USBTRG1 is automatically performed in the module. Therefore, do not write 1 to the EP1RDFN bit in USBTRG1 after reading the data on one side of the FIFO. If 1 is written to the EP1RDFN bit, correct operation cannot be guaranteed.

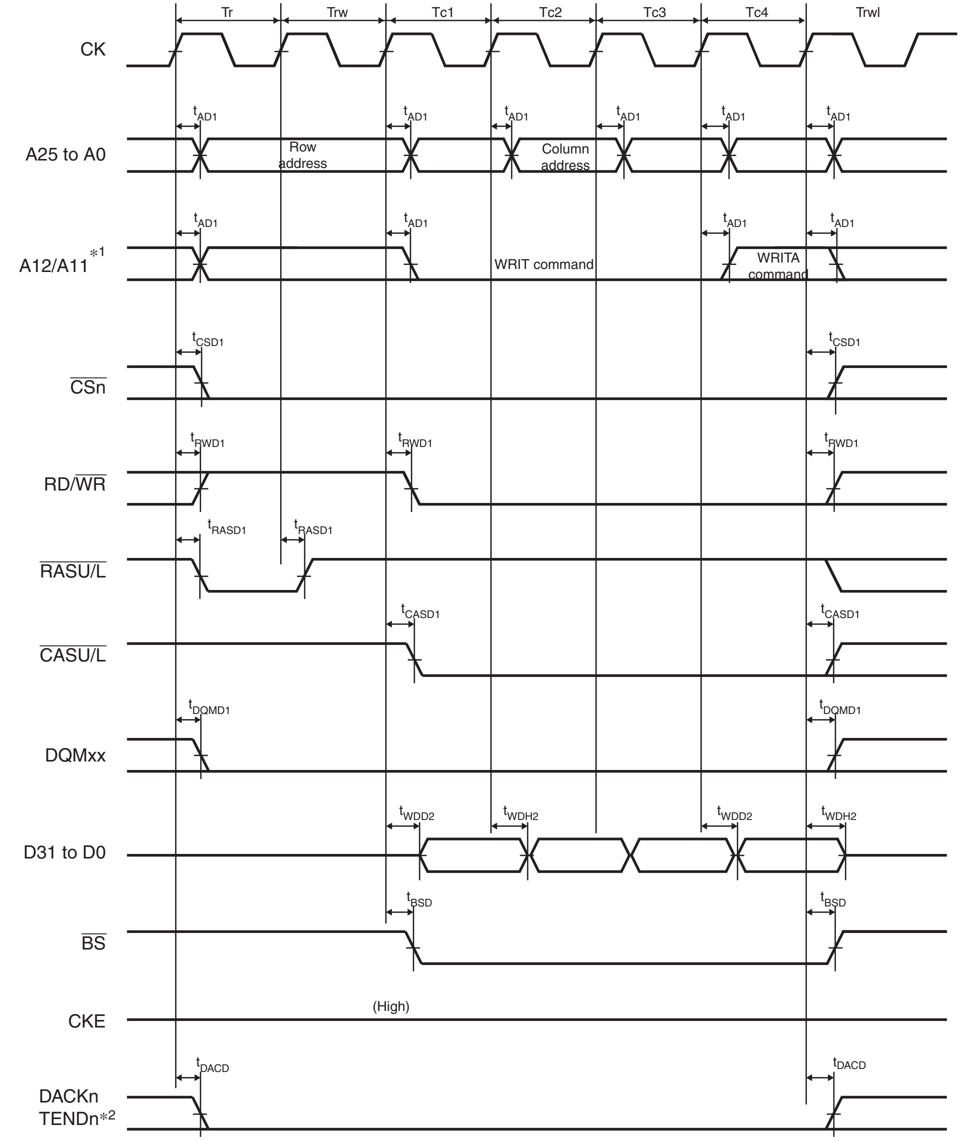
For example, if 150-byte data is received from the host, processing equivalent to writing 1 to the EP1RDFN bit in USBTRG1 is automatically performed internally in the three places in figure 24.22. Since this processing is performed when the data on the currently selected FIFO becomes empty, the processing is automatically performed in the same way even if data of 64 bytes or less is transferred.

Similarly, if the received data for EP4 is transferred by DTC, when the currently selected data FIFO becomes empty, processing equivalent to writing 1 to the EP4RDFN bit in USBTRG2 is automatically performed in the module. Therefore, do not write 1 to the EP4RDFN bit in USBTRG2 after reading the data on one side of the FIFO. If 1 is written to the EP4RDFN bit, correct operation cannot be guaranteed.

| No. | Pin Name | Type |
|---|---|---|
| 235 | PA20/$\overline{\text{WRL}}$/DQMLL/$\overline{\text{BREQ}}$/IRQ6/$\overline{\text{CASU}}$/$\overline{\text{POE4}}$/TXD1/$\overline{\text{AH}}$ | CONTROL |
| 234 | PA20/$\overline{\text{WRL}}$/DQMLL/$\overline{\text{BREQ}}$/IRQ6/$\overline{\text{CASU}}$/$\overline{\text{POE4}}$/TXD1/$\overline{\text{AH}}$ | INPUT |
| 233 | PA19/$\overline{\text{WRH}}$/DQMLU/$\overline{\text{WAIT}}$/IRQ7/$\overline{\text{RASU}}$/$\overline{\text{POE8}}$/RXD1/$\overline{\text{BS}}$ | OUTPUT |
| 232 | PA19/$\overline{\text{WRH}}$/DQMLU/$\overline{\text{WAIT}}$/IRQ7/$\overline{\text{RASU}}$/$\overline{\text{POE8}}$/RXD1/$\overline{\text{BS}}$ | CONTROL |
| 231 | PA19/$\overline{\text{WRH}}$/DQMLU/$\overline{\text{WAIT}}$/IRQ7/$\overline{\text{RASU}}$/$\overline{\text{POE8}}$/RXD1/$\overline{\text{BS}}$ | INPUT |
| 230 | PA18/CK | OUTPUT |
| 229 | PA18/CK | CONTROL |
| 228 | PA18/CK | INPUT |
| 227 | PA17/$\overline{\text{RD}}$ | OUTPUT |
| 226 | PA17/$\overline{\text{RD}}$ | CONTROL |
| 225 | PA17/$\overline{\text{RD}}$ | INPUT |
| 224 | PA16/$\overline{\text{WRL}}$/DQMLL | OUTPUT |
| 223 | PA16/$\overline{\text{WRL}}$/DQMLL | CONTROL |
| 222 | PA16/$\overline{\text{WRL}}$/DQMLL | INPUT |
| 221 | PA15/$\overline{\text{WRH}}$/DQMLU | OUTPUT |
| 220 | PA15/$\overline{\text{WRH}}$/DQMLU | CONTROL |
| 219 | PA15/$\overline{\text{WRH}}$/DQMLU | INPUT |
| 218 | PA14/$\overline{\text{WRHH}}$/DQMUU/$\overline{\text{RASL}}$ | OUTPUT |
| 217 | PA14/$\overline{\text{WRHH}}$/DQMUU/$\overline{\text{RASL}}$ | CONTROL |
| 216 | PA14/$\overline{\text{WRHH}}$/DQMUU/$\overline{\text{RASL}}$ | INPUT |
| 215 | PA13/$\overline{\text{WRHL}}$/DQMUL/$\overline{\text{CASL}}$ | OUTPUT |
| 214 | PA13/$\overline{\text{WRHL}}$/DQMUL/$\overline{\text{CASL}}$ | CONTROL |
| 213 | PA13/$\overline{\text{WRHL}}$/DQMUL/$\overline{\text{CASL}}$ | INPUT |
| 212 | PC0/A0/IRQ4/$\overline{\text{POE0}}$ | OUTPUT |
| 211 | PC0/A0/IRQ4/$\overline{\text{POE0}}$ | CONTROL |
| 210 | PC0/A0/IRQ4/$\overline{\text{POE0}}$ | INPUT |
| 209 | PC1/A1 | OUTPUT |
| 208 | PC1/A1 | CONTROL |
| 207 | PC1/A1 | INPUT |
| 206 | PC2/A2 | OUTPUT |
| 205 | PC2/A2 | CONTROL |
| 204 | PC2/A2 | INPUT |

| Module Name | Register Name | Abbreviation | Number of Bits | Address | Access Size |
|---|---|---|---|---|---|
| PFC | Port D pull-up MOS control register H | PDPCRH | 16 | H'FFFE39A8 | 8, 16, 32 |
| | Port D pull-up MOS control register L | PDPCRL | 16 | H'FFFE39AA | 8, 16 |
| | Port E IO register L | PEIORL | 16 | H'FFFE3A06 | 8, 16 |
| | Port E control register L4 | PECRL4 | 16 | H'FFFE3A10 | 8, 16, 32 |
| | Port E control register L3 | PECRL3 | 16 | H'FFFE3A12 | 8, 16 |
| | Port E control register L2 | PECRL2 | 16 | H'FFFE3A14 | 8, 16, 32 |
| | Port E control register L1 | PECRL1 | 16 | H'FFFE3A16 | 8, 16 |
| | Large current port control register | HCPCR | 16 | H'FFFE3A20 | 8, 16, 32 |
| | IRQOUT function control register | IFCR | 16 | H'FFFE3A22 | 8, 16 |
| | Port E pull-up MOS control register L | PEPCRL | 16 | H'FFFE3A2A | 8, 16 |
| | DACK output timing control register | PDACKCR | 16 | H'FFFE3A2C | 8, 16 |
| I/O port | Port A data register H | PADRH | 16 | H'FFFE3800 | 8, 16, 32 |
| | Port A data register L | PADRL | 16 | H'FFFE3802 | 8, 16 |
| | Port A port register H | PAPRH | 16 | H'FFFE381C | 8, 16, 32 |
| | Port A port register L | PAPRL | 16 | H'FFFE381E | 8, 16 |
| | Port B data register L | PBDRL | 16 | H'FFFE3882 | 8, 16 |
| | Port B port register L | PBPRL | 16 | H'FFFE389E | 8, 16 |
| | Port C data register L | PCDRL | 16 | H'FFFE3902 | 8, 16 |
| | Port C port register L | PCPRL | 16 | H'FFFE391E | 8, 16 |
| | Port D data register H | PDDRH | 16 | H'FFFE3980 | 8, 16, 32 |
| | Port D data register L | PDDRL | 16 | H'FFFE3982 | 8, 16 |
| | Port D port register H | PDPRH | 16 | H'FFFE399C | 8, 16, 32 |
| | Port D port register L | PDPRL | 16 | H'FFFE399E | 8, 16 |
| | Port E data register L | PEDRL | 16 | H'FFFE3A02 | 8, 16 |
| | Port E port register L | PEPRL | 16 | H'FFFE3A1E | 8, 16 |
| | Port F data register L | PFDRL | 16 | H'FFFE3A82 | 8, 16 |
| USB | USB interrupt flag register 0 | USBIFR0 | 8 | H'FFFE7000 | 8 |
| | USB interrupt flag register 1 | USBIFR1 | 8 | H'FFFE7001 | 8 |
| | USB interrupt flag register 2 | USBIFR2 | 8 | H'FFFE7002 | 8 |
| | USB interrupt flag register 3 | USBIFR3 | 8 | H'FFFE7003 | 8 |
| | USB interrupt flag register 4 | USBIFR4 | 8 | H'FFFE7004 | 8 |
| | USB interrupt enable register 0 | USBIER0 | 8 | H'FFFE7008 | 8 |

| Module Name | Register | Power-on Reset | Manual Reset | Software Standby | Module Standby | Sleep |
|---|---|---|---|---|---|---|
| BSC | CS1WCR | Initialized | Retained | Retained | — | Retained |
| | CS2WCR | Initialized | Retained | Retained | — | Retained |
| | CS3WCR | Initialized | Retained | Retained | — | Retained |
| | CS4WCR | Initialized | Retained | Retained | — | Retained |
| | CS5WCR | Initialized | Retained | Retained | — | Retained |
| | CS6WCR | Initialized | Retained | Retained | — | Retained |
| | CS7WCR | Initialized | Retained | Retained | — | Retained |
| | SDCR | Initialized | Retained | Retained | — | Retained |
| | RTCSR | Initialized | Retained (Flag processing continued) | Retained | — | Retained (Flag processing continued) |
| | RTCNT | Initialized | Retained (Count-up continued) | Retained | — | Retained (Count-up continued) |
| | RTCOR | Initialized | Retained | Retained | — | Retained |
| | BSCEHR | Initialized | Retained | Retained | — | Retained |
| DMAC | SAR_0 | Initialized | Retained | Retained | Retained | Retained |
| | DAR_0 | Initialized | Retained | Retained | Retained | Retained |
| | DMATCR_0 | Initialized | Retained | Retained | Retained | Retained |
| | CHCR_0 | Initialized | Retained | Retained | Retained | Retained |
| | RSAR_0 | Initialized | Retained | Retained | Retained | Retained |
| | RDAR_0 | Initialized | Retained | Retained | Retained | Retained |
| | RDMATCR_0 | Initialized | Retained | Retained | Retained | Retained |
| | SAR_1 | Initialized | Retained | Retained | Retained | Retained |
| | DAR_1 | Initialized | Retained | Retained | Retained | Retained |
| | DMATCR_1 | Initialized | Retained | Retained | Retained | Retained |
| | CHCR_1 | Initialized | Retained | Retained | Retained | Retained |
| | RSAR_1 | Initialized | Retained | Retained | Retained | Retained |
| | RDAR_1 | Initialized | Retained | Retained | Retained | Retained |
| | RDMATCR_1 | Initialized | Retained | Retained | Retained | Retained |



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

**Figure 33.25 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)**

### 33.3.12 RSPI Timing

**Table 33.16 SPI Timing**

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to $3.6$ V, $AV_{CC} = AVREF = 4.5$ to $5.5$V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFV_{SS} = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

| Item | | Symbol | Min. | Typ. | Max. | Unit | Figure |
|---|---|---|---|---|---|---|---|
| RSPCK clock cycle*[1] | Master | $t_{SPcyc}$ | 2 | — | 4096 | $t_{Pcyc}$ | Figure 33.49 |
| | Slave | | 8 | — | 4096 | | |
| RSPCK clock cycle high pulse width | Master | $t_{SPCKWH}$ | $(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2 - 3$ | — | — | ns | |
| | Slave | | $(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2$ | — | — | | |
| RSPCK clock cycle low pulse width | Master | $t_{SPCKWL}$ | $(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2 - 3$ | — | — | ns | |
| | Slave | | $(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2$ | — | — | | |
| RSPCK clock rise/fall time*[2] | Master | $t_{SPCKR}$, $t_{SPCKF}$ | — | — | 5 | ns | |
| | Slave | | — | — | 1 | $t_{Pcyc}$ | |
| Data input setup time | Master | $t_{SU}$ | 25 | — | — | ns | Figures 33.50 to 33.53 |
| | Slave | | $20 - 2 \times t_{Pcyc}$ | — | — | | |
| Data input hold time | Master | $t_{H}$ | 0 | — | — | ns | |
| | Slave | | $20 + 2 \times t_{Pcyc}$ | — | — | | |
| SSL setup time | Master | $t_{LEAD}$ | 1 | — | 8 | $t_{SPcyc}$ | |
| | Slave | | 4 | — | — | $t_{Pcyc}$ | |
| SSL hold time | Master | $t_{LAG}$ | 1 | — | 8 | $t_{SPcyc}$ | |
| | Slave | | 4 | — | — | $t_{Pcyc}$ | |
| Data output delay time | Master | $t_{OD}$ | — | — | 10 | ns | |
| | Slave | | — | — | $3 \times t_{Pcyc} + 15$ | | |
| Data output hold time | Master | $t_{OH}$ | 0 | — | — | ns | |
| | Slave | | 0 | — | — | | |
| Continuous transmission delay time | Master | $t_{TD}$ | $t_{SPcyc} + 2 \times t_{Pcyc}$ | — | $8 \times t_{SPcyc} + 2 \times t_{Pcyc}$ | ns | |
| | Slave | | $4 \times t_{Pcyc}$ | — | — | | |