# Renesas Electronics America Inc - R5F61668MZN50FPV Datasheet

**Welcome to E-XFL.COM**

## 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 | H8SX |
| Core Size | 32-Bit Single-Core |
| Speed | 50MHz |
| Connectivity | EBI/EMI, I²C, IrDA, SCI, SmartCard, USB |
| Peripherals | DMA, LVD, POR, PWM, WDT |
| Number of I/O | 92 |
| Program Memory Size | 1MB (1M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 56K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 8x10b; D/A 2x8b |
| Oscillator Type | External |
| Operating Temperature | -20°C ~ 75°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 144-LQFP |
| Supplier Device Package | 144-LFQFP (20x20) |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f61668mzn50fpv |

**Table 2.2 Combinations of Instructions and Addressing Modes (2)**

| | | | Addressing Mode | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Classification | Instruction | Size | @ERn | @(d,PC) | @(RnL.B/Rn.W/ERn.L, PC) | @aa:24 | @aa:32 | @@ aa:8 | @@vec:7 | — |
| Branch | BRA/BS, BRA/BC | — | | O | | | | | | |
| | BSR/BS, BSR/BC | — | | O | | | | | | |
| | Bcc | — | | O | | | | | | |
| | BRA | — | | O | O | | | | | |
| | BRA/S | — | | O* | | | | | | |
| | JMP | — | O | | | O | O | O | O | |
| | BSR | — | | O | | | | | | |
| | JSR | — | O | | | O | O | O | O | |
| | RTS, RTS/L | — | | | | | | | | O |
| System control | TRAPA | — | | | | | | | | O |
| | RTE, RTE/L | — | | | | | | | | O |

[Legend]

d:d:8 or d:16

Note: * Only @(d:8, PC) is available.

### 2.8.2 Register Indirect—@ERn

The operand value is the contents of the memory location which is pointed to by the contents of an address register (ERn). ERn is specified by the register field of the instruction code.

In advanced mode, if this addressing mode is used in a branch instruction, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

### 2.8.3 Register Indirect with Displacement —@(d:2, ERn), @(d:16, ERn), or @(d:32, ERn)

The operand value is the contents of a memory location which is pointed to by the sum of the contents of an address register (ERn) and a 16- or 32-bit displacement. ERn is specified by the register field of the instruction code. The displacement is included in the instruction code and the 16-bit displacement is sign-extended when added to ERn.

This addressing mode has a short format (@(d:2, ERn)). The short format can be used when the displacement is 1, 2, or 3 and the operand is byte data, when the displacement is 2, 4, or 6 and the operand is word data, or when the displacement is 4, 8, or 12 and the operand is longword data.

### 2.8.4 Index Register Indirect with Displacement—@(d:16,RnL.B), @(d:32,RnL.B), @(d:16,Rn.W), @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)

The operand value is the contents of a memory location which is pointed to by the sum of the following operation result and a 16- or 32-bit displacement: a specified bits of the contents of an address register (RnL, Rn, ERn) specified by the register field in the instruction code are zero-extended to 32-bit data and multiplied by 1, 2, or 4. The displacement is included in the instruction code and the 16-bit displacement is sign-extended when added to ERn. If the operand is byte data, ERn is multiplied by 1. If the operand is word or longword data, ERn is multiplied by 2 or 4, respectively.

## 6.5 Address Error

### 6.5.1 Address Error Source

Instruction fetch, stack operation, or data read/write shown in table 6.5 may cause an address error.

**Table 6.5 Bus Cycle and Address Error**

| | Bus Cycle | | |
|---|---|---|---|
| **Type** | **Bus Master** | **Description** | **Address Error** |
| Instruction fetch | CPU | Fetches instructions from even addresses | No (normal) |
| | | Fetches instructions from odd addresses | Occurs |
| | | Fetches instructions from areas other than on-chip peripheral module space*[1] | No (normal) |
| | | Fetches instructions from on-chip peripheral module space*[1] | Occurs |
| | | Fetches instructions from external memory space in single-chip mode | Occurs |
| | | Fetches instructions from access prohibited area.*[2] | Occurs |
| Stack operation | CPU | Accesses stack when the stack pointer value is even address | No (normal) |
| | | Accesses stack when the stack pointer value is odd address | Occurs |
| Data read/write | CPU | Accesses word data from even addresses | No (normal) |
| | | Accesses word data from odd addresses | No (normal) |
| | | Accesses external memory space in single-chip mode | Occurs |
| | | Accesses to access prohibited area*[2] | Occurs |
| Data read/write | DTC or DMAC | Accesses word data from even addresses | No (normal) |
| | | Accesses word data from odd addresses | No (normal) |
| | | Accesses external memory space in single-chip mode | Occurs |
| | | Accesses to access prohibited area*[2] | Occurs |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | W52 | 1 | R/W | Area 5 Wait Control 2 to 0 |
| 5 | W51 | 1 | R/W | These bits select the number of program wait cycles when accessing area 5 while bit AST5 in ASTCR is 1. |
| 4 | W50 | 1 | R/W | 000: Program cycle wait not inserted |
| | | | | 001: 1 program wait cycle inserted |
| | | | | 010: 2 program wait cycles inserted |
| | | | | 011: 3 program wait cycles inserted |
| | | | | 100: 4 program wait cycles inserted |
| | | | | 101: 5 program wait cycles inserted |
| | | | | 110: 6 program wait cycles inserted |
| | | | | 111: 7 program wait cycles inserted |
| 3 | — | 0 | R | Reserved |
| | | | | This is a read-only bit and cannot be modified. |
| 2 | W42 | 1 | R/W | Area 4 Wait Control 2 to 0 |
| 1 | W41 | 1 | R/W | These bits select the number of program wait cycles when accessing area 4 while bit AST4 in ASTCR is 1. |
| 0 | W40 | 1 | R/W | 000: Program wait cycle not inserted |
| | | | | 001: 1 program wait cycle inserted |
| | | | | 010: 2 program wait cycles inserted |
| | | | | 011: 3 program wait cycles inserted |
| | | | | 100: 4 program wait cycles inserted |
| | | | | 101: 5 program wait cycles inserted |
| | | | | 110: 6 program wait cycles inserted |
| | | | | 111: 7 program wait cycles inserted |

- WTCRB

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved |
| | | | | This is a read-only bit and cannot be modified. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | W32 | 1 | R/W | Area 3 Wait Control 2 to 0 |
| 13 | W31 | 1 | R/W | These bits select the number of program wait cycles when accessing area 3 while bit AST3 in ASTCR is 1. |
| 12 | W30 | 1 | R/W | 000: Program wait cycle not inserted<br>001: 1 program wait cycle inserted<br>010: 2 program wait cycles inserted<br>011: 3 program wait cycles inserted<br>100: 4 program wait cycles inserted<br>101: 5 program wait cycles inserted<br>110: 6 program wait cycles inserted<br>111: 7 program wait cycles inserted |
| 11 | — | 0 | R | Reserved<br>This is a read-only bit and cannot be modified. |
| 10 | W22 | 1 | R/W | Area 2 Wait Control 2 to 0 |
| 9 | W21 | 1 | R/W | These bits select the number of program wait cycles when accessing area 2 while bit AST2 in ASTCR is 1. When SDRAM is connected, the CAS latency is specified. At this time, W22 is ignored. The CAS latency can be specified even if the wait cycle insertion is disabled by ASTCR. |
| 8 | W20 | 1 | R/W | Selection of number of program wait cycles:<br>000: Program wait cycle not inserted<br>001: 1 program wait cycle inserted<br>010: 2 program wait cycles inserted<br>011: 3 program wait cycles inserted<br>100: 4 program wait cycles inserted<br>101: 5 program wait cycles inserted<br>110: 6 program wait cycles inserted<br>111: 7 program wait cycles inserted<br>Setting of CAS latency (W22 is ignored.):<br>00: Setting prohibited<br>01: SDRAM with a CAS latency of 2 is connected.<br>10: SDRAM with a CAS latency of 3 is connected.<br>11: SDRAM with a CAS latency of 4 is connected. |

### 9.2.10 SRAM Mode Control Register (SRAMCR)

SRAMCR specifies the bus interface of each area in the external address space as a basic bus interface or a byte control SRAM interface.

In areas specified as 8-bit access space by ABWCR, the SRAMCR setting is ignored and the byte control SRAM interface cannot be specified.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Bit Name | BCSEL7 | BCSEL6 | BCSEL5 | BCSEL4 | BCSEL3 | BCSEL2 | BCSEL1 | BCSEL0 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Name | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | BCSEL7 | 0 | R/W | Byte Control SRAM Interface Select |
| 14 | BCSEL6 | 0 | R/W | Selects the bus interface for the corresponding area. |
| 13 | BCSEL5 | 0 | R/W | When setting a bit to 1, the bus interface select bits in BROMCR, DRAMCR and MPXCR must be cleared to 0. |
| 12 | BCSEL4 | 0 | R/W | |
| 11 | BCSEL3 | 0 | R/W | 0: Area n is basic bus interface |
| 10 | BCSEL2 | 0 | R/W | 1: Area n is byte control SRAM interface |
| 9 | BCSEL1 | 0 | R/W | (n = 7 to 0) |
| 8 | BCSEL0 | 0 | R/W | |
| 7 to 0 | — | All 0 | R | Reserved<br>These are read-only bits and cannot be modified. |

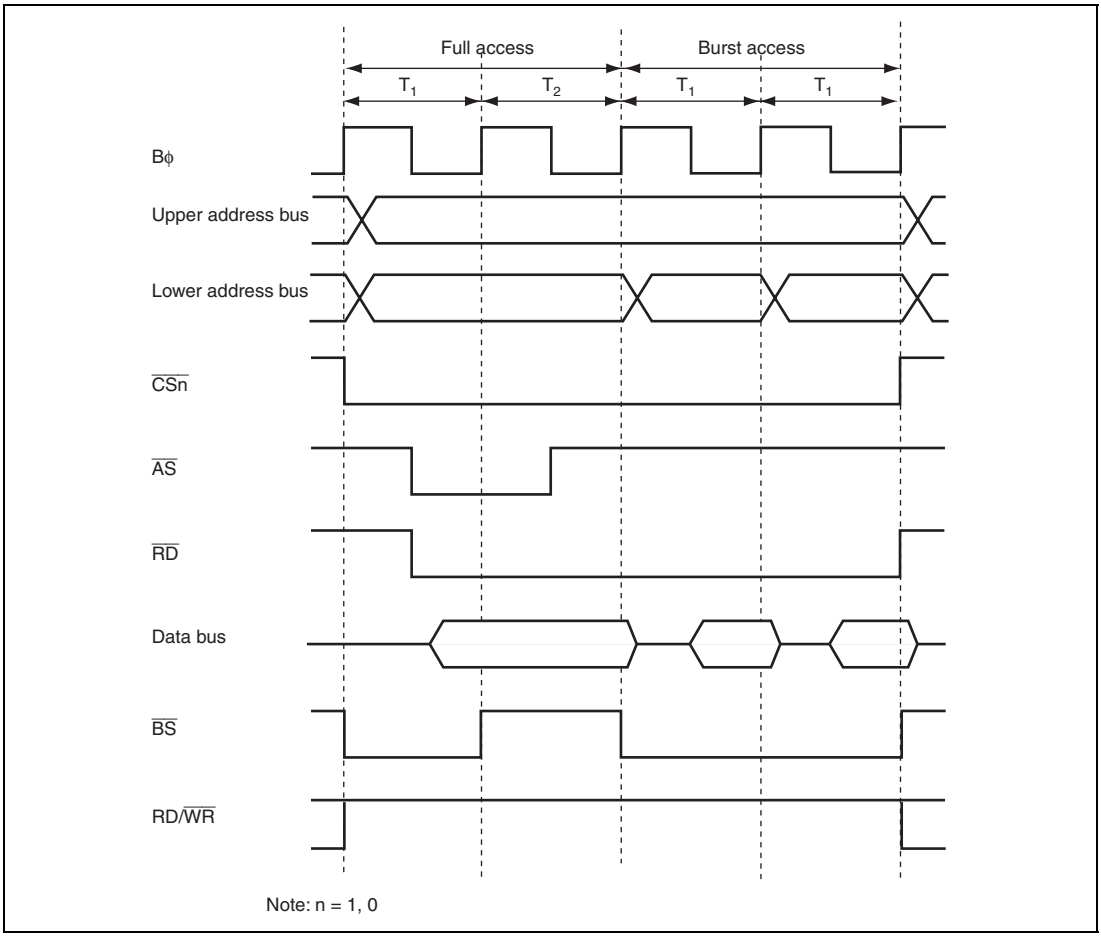



**Figure 9.30 Example of Burst ROM Access Timing (ASTn = 0, One Burst Cycle)**

### (3) Refresh and All-Module Clock Stop Mode

This LSI is entered in all-module clock stop mode by the following operation: Stop the clocks of all on-chip peripheral modules by setting the ACSE bit in MSTPCR to 1 (MSTPCRA, MSTPCRB = H'FFFFFFFF) or run only the 8-bit timer (MSTPCRA, MSTPCRB = H'F[C to F]FFFFFF), then execute the SLEEP instruction to enter the sleep mode.

In all-module clock stop mode, clocks for the bus controller and I/O ports are stopped. Since the clock for the bus controller is stopped, a CBR refresh cycle cannot be performed. When external DRAM is used and the contents of the DRAM in sleep mode should be held, clear the ACSE bit in MSTPCE to 0.

For details, see section 28.2.2, Module Stop Control Registers A and B (MSTPCRA and MSTPCRB).

## 9.10.13 DRAM Interface and Single Address Transfer by DMAC and EXDMAC

When fast-page mode (BE = 1) is set for the DRAM space, either fast-page access or full access can be selected, by the setting of bits DDS and EDDS in DRAMCR, for the single address transfer by the DMAC or EXDMAC where the DRAM space is specified as the transfer source or destination. At the same time, the output timings of the $\overline{\text{DACK}}$, $\overline{\text{EDACK}}$ and BS signals are changed. When BE = 0, full access to the DRAM space is performed by single address transfer regardless of the setting of bits DDS and EDDS. However, the output timing of the $\overline{\text{DACK}}$, $\overline{\text{EDACK}}$ and BS signals can be changed by the setting of bits DDS and EDDS.

The assertion timing of the $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ signal can be changed by bits DKC and EDKC in BCR1.

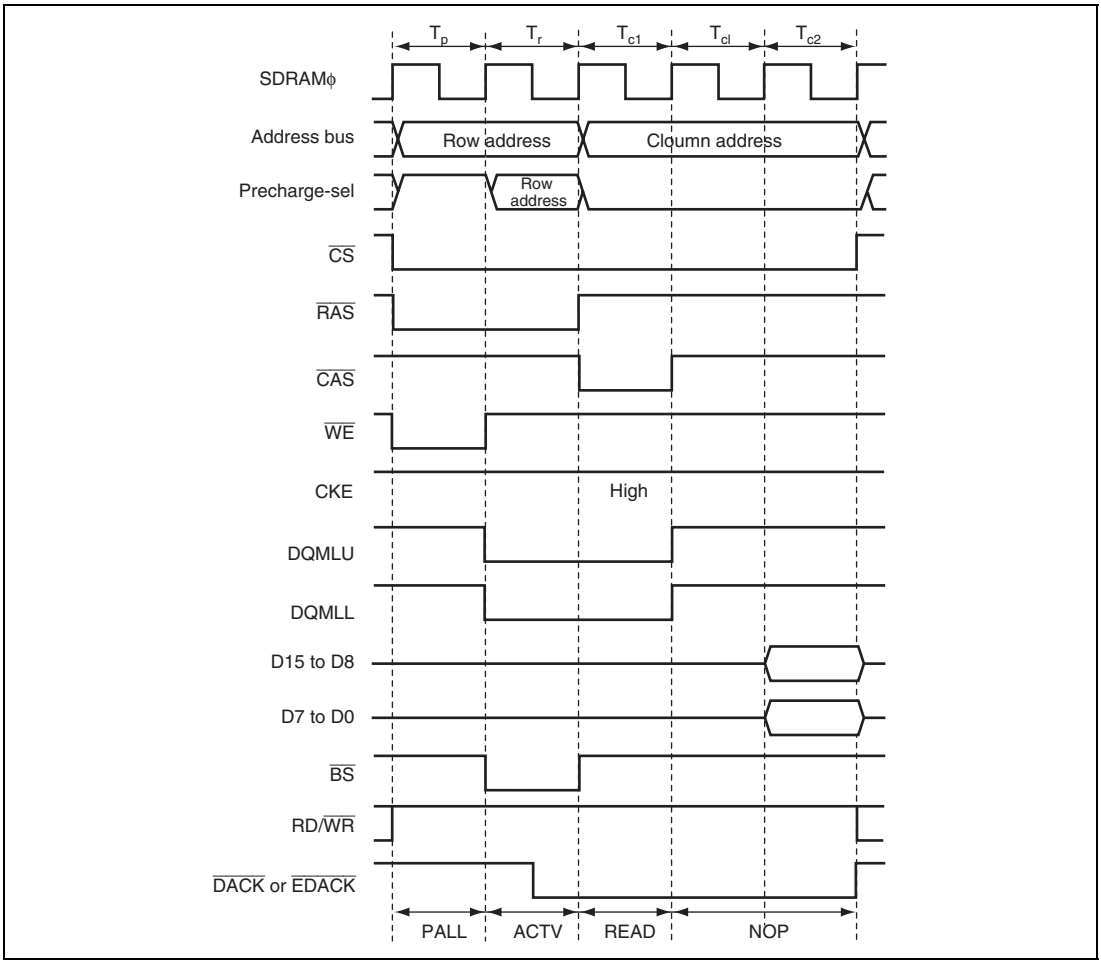


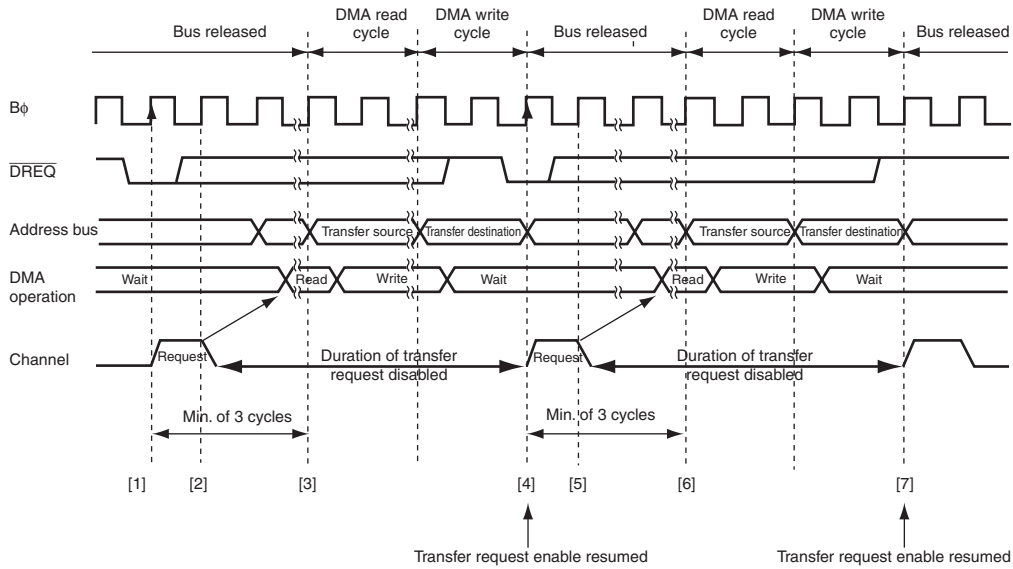

**Figure 9.86 Output Timing Example of $\overline{\text{DACK}}$ and $\overline{\text{EDACK}}$ when DDS = 0 or EDDS = 0 (Read, CAS Latency = 2)**

### (4) Activation Timing by $\overline{\text{DREQ}}$ Falling Edge

Figure 10.29 shows an example of normal transfer mode activated by the $\overline{\text{DREQ}}$ signal falling edge.

The $\overline{\text{DREQ}}$ signal is sampled every cycle from the next rising edge of the Bϕ signal immediately after the DTE bit write cycle.

When a low level of the $\overline{\text{DREQ}}$ signal is detected while a transfer request by the $\overline{\text{DREQ}}$ signal is enabled, a transfer request is held in the DMAC. When the DMAC is activated, the transfer request is cleared and starts detecting a high level of the $\overline{\text{DREQ}}$ signal for falling edge detection. If a high level of the $\overline{\text{DREQ}}$ signal has been detected until completion of the DMA write cycle, receiving the next transfer request resumes and then a low level of the $\overline{\text{DREQ}}$ signal is detected. This operation is repeated until the transfer is completed.




[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started and sampling the $\overline{\text{DREQ}}$ signal at the rising edge of the Bϕ signal is started to detect a high level of the $\overline{\text{DREQ}}$ signal.
[4][7] When a high level of the $\overline{\text{DREQ}}$ signal has been detected, transfer request enable is resumed after completion of the write cycle. (A low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the same as [1].)

**Figure 10.29 Example of Transfer in Normal Transfer Mode Activated by $\overline{\text{DREQ}}$ Falling Edge**

### (3) Block Transfer Mode

In block transfer mode, one block is transferred in response to one transfer request, and after the transfer, the bus is released.

Figure 11.28 shows an example of transfer when $\overline{\text{ETEND}}$ output is enabled, and word-size, block transfer mode is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

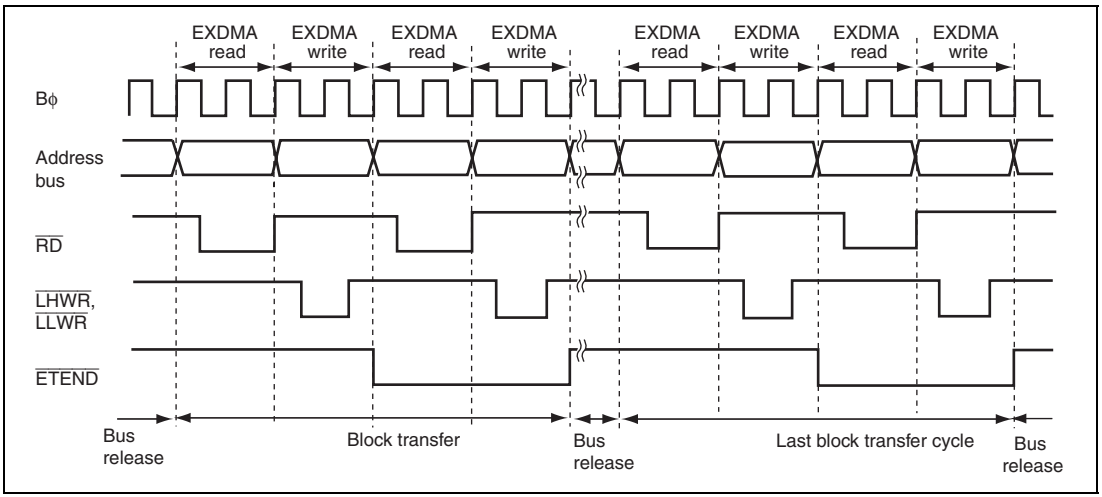



**Figure 11.28 Example of Block Transfer Mode Transfer**

### 15.4.3 Example of Normal Pulse Output (Example of 5-Phase Pulse Output)

Figure 15.5 shows an example in which pulse output is used for cyclic 5-phase pulse output.

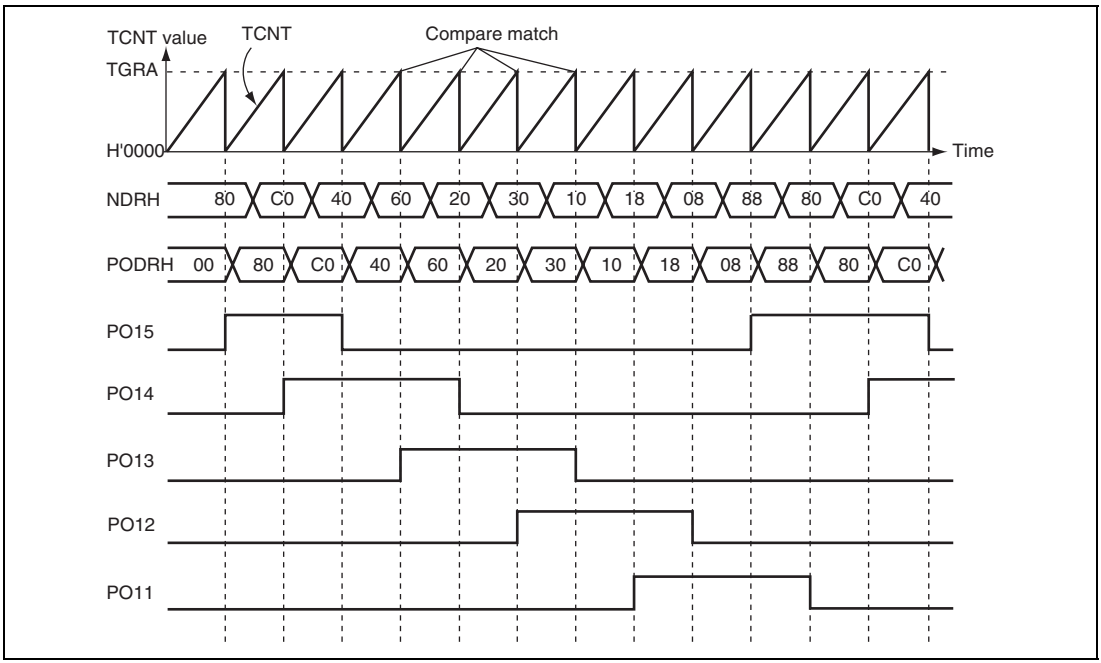
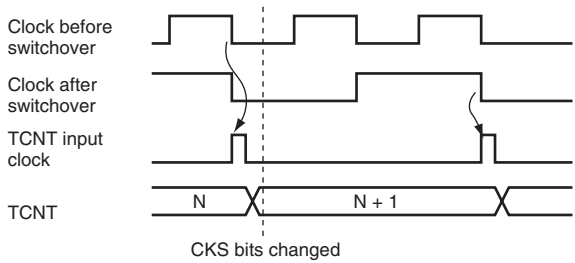
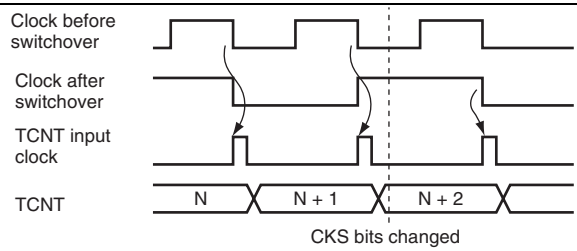
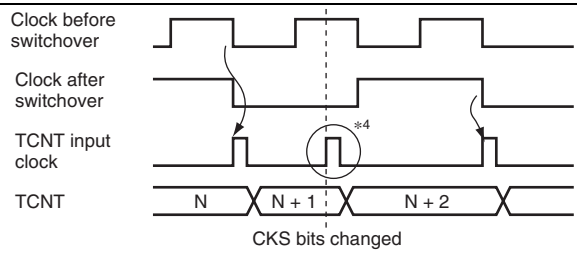
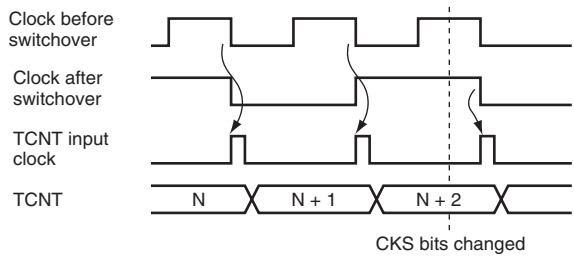



**Figure 15.5 Normal Pulse Output Example (5-Phase Pulse Output)**

1. Set up TGRA in TPU which is used as the output trigger to be an output compare register. Set a cycle in TGRA so the counter will be cleared by compare match A. Set the TGIEA bit in TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 to NDERH, and set bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.
4. 5-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupts.

   If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

**Table 16.9 Switching of Internal Clock and TCNT Operation**

| No. | Timing to Change CKS1 and CKS0 Bits | TCNT Clock Operation |
|---|---|---|
| 1 | Switching from low to low*[1] |  |
| 2 | Switching from low to high*[2] |  |
| 3 | Switching from high to low*[3] |  |
| 4 | Switching from high to high |  |

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated because the change of the signal levels is considered as a falling edge; TCNT is incremented.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | CKE1 | 0 | R/W | Clock Enable 1, 0 (for SCI_6) |
| 0 | CKE0 | 0 | R/W | These bits select the clock source.<br>• Asynchronous mode<br>00: On-chip baud rate generator<br>1X: TMR clock input or average transfer rate generator<br>When an average transfer rate generator is used.<br>When TMR clock input is used.<br>• Clocked synchronous mode<br>Not available |

[Legend]

X: Don't care

## 19.7 Operation in Smart Card Interface Mode

The SCI supports the smart card interface, supporting the ISO/IEC 7816-3 (Identification Card) standard, as an extended serial communication interface function. Smart card interface mode can be selected using the appropriate register.

### 19.7.1 Sample Connection

Figure 19.24 shows a sample connection between the smart card and this LSI. As in the figure, since this LSI communicates with the smart card using a single transmission line, interconnect the TxD and RxD pins and pull up the data transmission line to $V_{CC}$ using a resistor. Setting the RE and TE bits to 1 with the smart card not connected enables closed transmission/reception allowing self diagnosis. To supply the smart card with the clock pulses generated by the SCI, input the SCK pin output to the CLK pin of the smart card. A reset signal can be supplied via the output port of this LSI. (In SCI_6, the clock generated in SCI cannot be provided to smart cards.)

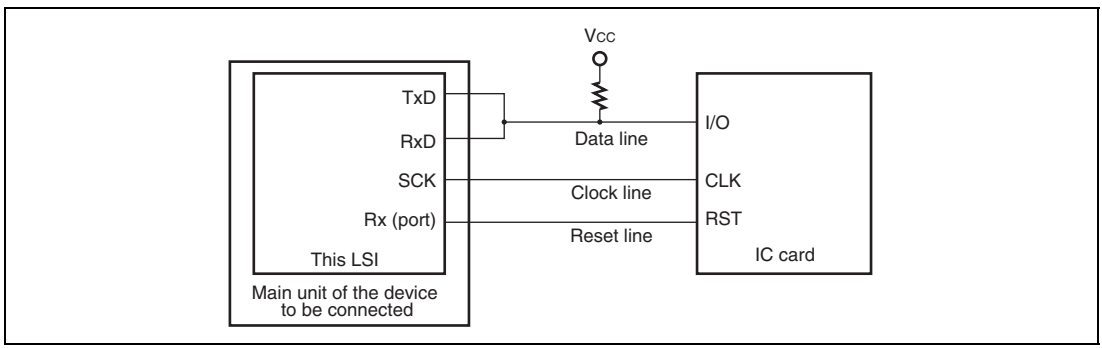



**Figure 19.24 Pin Connection for Smart Card Interface**

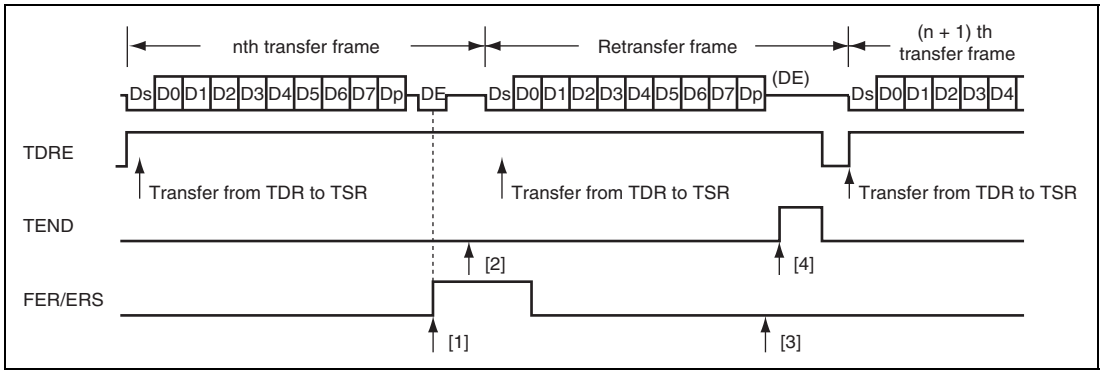



**Figure 19.29 Data Re-Transfer Operation in SCI Transmission Mode**

Note that the TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 19.30 shows the TEND flag set timing.

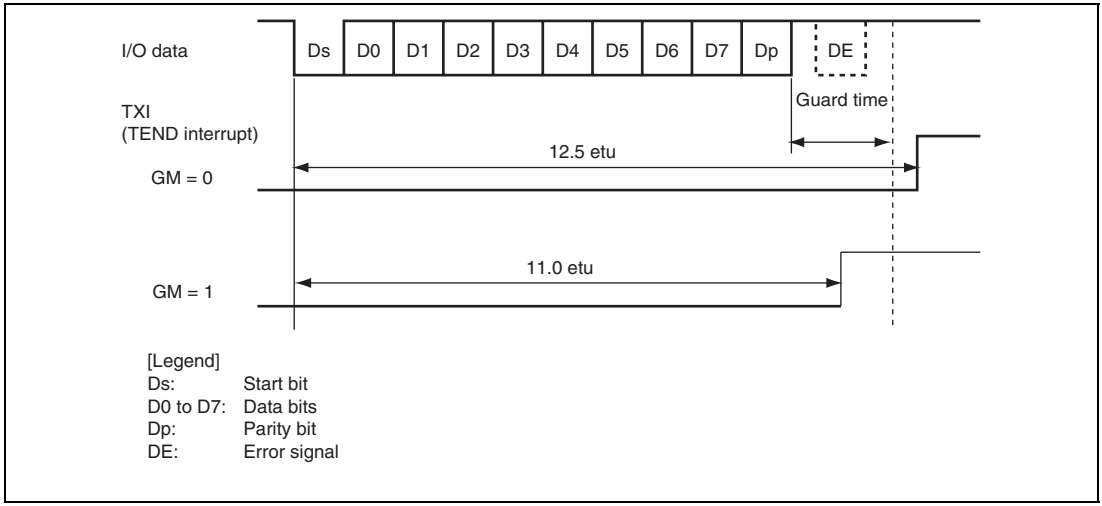



**Figure 19.30 TEND Flag Set Timing during Transmission**

### 20.3.6 Interrupt Select Register 2 (ISR2)

ISR2 selects the vector numbers of the interrupt requests indicated in interrupt flag register 2 (IFR2). If the USB issues an interrupt request to the INTC when a bit in ISR2 is cleared to 0, the interrupt corresponding to the bit will be USBINTN2. If the USB issues an interrupt request to the INTC when a bit in ISR2 is set to 1, the corresponding interrupt will be USBINTN3.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Name | — | — | — | SURSE | CFDN | — | SETCE | SETIE |
| Initial Value | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R/W | R/W | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved |
| 6 | — | 0 | R | These bits are always read as 0. The write value should always be 0. |
| 5 | — | 0 | R | |
| 4 | SURSE | 1 | R/W | Suspend/Resume Detection |
| 3 | CFDN | 1 | R/W | End Point Information Load End |
| 2 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 1 | SETCE | 1 | R/W | Set_Configuration Command Detection |
| 0 | SETIE | 1 | R/W | Set_Interface Command Detection |

### 21.3.3 I²C Bus Mode Register (ICMR)

ICMR selects MSB first or LSB first, controls the master mode wait and selects the number of transfer bits.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Name | — | WAIT | — | — | BCWP | BC2 | BC1 | BC0 |
| Initial Value | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| R/W | R/W | R/W | — | — | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R/W | Reserved<br>The write value should always be 0. |
| 6 | WAIT | 0 | R/W | Wait Insertion<br>This bit selects whether to insert a wait after data transfer except for the acknowledge bit. When this bit is set to 1, after the falling of the clock for the last data bit, the low period is extended for two transfer clocks. When this bit is cleared to 0, data and the acknowledge bit are transferred consecutively with no waits inserted. The setting of this bit is invalid in slave mode. |
| 5 | — | 1 | — | Reserved |
| 4 | — | 1 | — | These bits are always read as 1. |
| 3 | BCWP | 1 | R/W | BC Write Protect<br>This bit controls the modification of the BC2 to BC0 bits. When modifying, this bit should be cleared to 0 and the MOV instruction should be used.<br>0: When writing, the values of BC2 to BC0 are set<br>1: When reading, 1 is always read<br>When writing, the settings of BC2 to BC0 are invalid. |

### (k) New Bit-Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This selection should be sent after sending the clock mode selection command.

| Command | H'3F | Size | Bit rate | Input frequency |
|---|---|---|---|---|
| | Number of types of multiplication | Multiplication ratio 1 | Multiplication ratio 2 | |
| | SUM | | | |

- Command, H'3F, (one byte): Selection of new bit rate
- Size (one byte): The number of bytes that represents the bit rate, input frequency, number of types of multiplication, and multiplication ratio
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is H'00C0.)
- Input frequency (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 100. (E.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Number of types of multiplication (one byte): The number of multiplication to which the device can be set.
- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for the main operating frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)
  Division ratio: The inverse of the division ratio, as a negative number (e.g. when the clock frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for the peripheral frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)
  (Division ratio: The inverse of the division ratio, as a negative number (E.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- SUM (one byte): Checksum

| Response | H'06 |
|---|---|

- Response, H'06, (one byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

## 28.12 Usage Notes

### 28.12.1 I/O Port Status

In software standby mode or deep software standby mode, the I/O port states are retained. Therefore, there is no reduction in current drawn due to output currents when high-level signals are being output.

### 28.12.2 Current Consumption during Oscillation Settling Standby Period

Current consumption increases during the oscillation settling standby period.

### 28.12.3 Module Stop State of EXDMAC, DMAC, or DTC

Depending on the operating state of the EXDMAC, DMAC, and DTC, bits MSTPA14, MSTPA13, and MSTPA12 may not be set to 1, respectively. The module stop state setting for the EXDMAC, DMAC, or DTC should be carried out only when the EXDMAC, DMAC, or DTC is not activated.

For details, refer to section 10, DMA Controller (DMAC), section 11, EXDMA Controller (EXDMAC), and section 12, Data Transfer Controller (DTC).

### 28.12.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in a module stop state. Consequently, if the module stop state is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts should therefore be disabled before entering a module stop state.

### 28.12.5 Writing to MSTPCRA, MSTPCRB, and MSTPCRC

MSTPCRA, MSTPCRB, and MSTPCRC should only be written to by the CPU.