# Renesas Electronics America Inc - R5F61663MN50FPV 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 | Obsolete |
| 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 | 384KB (384K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 40K 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/r5f61663mn50fpv |

### 9.5.2 Area Division

The bus controller divides the 16-Mbyte address space into eight areas, and performs bus control for the external address space in area units. Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for each area.

Figure 9.8 shows an area division of the 16-Mbyte address space. For details on address map, see section 3, MCU Operating Modes.

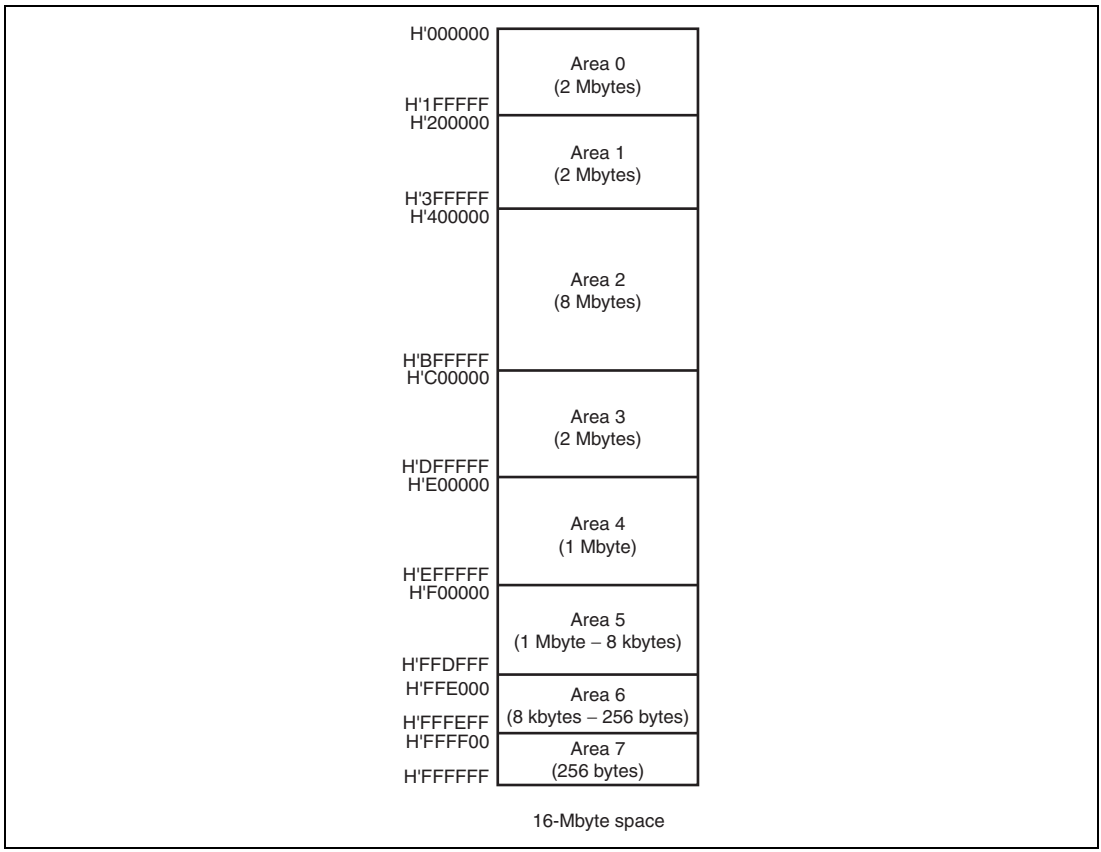



**Figure 9.8 Address Space Area Division**

### (4) Number of Access Cycles

#### (a) Basic Bus Interface

The number of access cycles in the basic bus interface can be specified as two or three cycles by the ASTCR. An area specified as 2-state access is specified as 2-state access space; an area specified as 3-state access is specified as 3-state access space.

For the 2-state access space, a wait cycle insertion is disabled. For the 3-state access space, a program wait (0 to 7 cycles) specified by WTCRA and WTCRB or an external wait by $\overline{\text{WAIT}}$ can be inserted.

Assertion period of the chip select signal can be extended by CSACR.

Number of access cycles in the basic bus interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{\text{CS}}$ extension cycles (0, 1, 2)
[+ number of external wait cycles by the $\overline{\text{WAIT}}$ pin]

#### (b) Byte Control SRAM Interface

The number of access cycles in the byte control SRAM interface is the same as that in the basic bus interface.

Number of access cycles in byte control SRAM interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{\text{CS}}$ extension cycles (0, 1, 2)
[+ number of external wait cycles by the $\overline{\text{WAIT}}$ pin]

#### (c) Burst ROM Interface

The number of access cycles at full access in the burst ROM interface is the same as that in the basic bus interface. The number of access cycles in the burst access can be specified as one to eight cycles by the BSTS bit in BROMCR.

Number of access cycles in the burst ROM interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{\text{CS}}$ extension cycles (0, 1)
[+number of external wait cycles by the $\overline{\text{WAIT}}$ pin]
+ number of burst access cycles (1 to 8) $\times$ number of burst accesses (0 to 63)

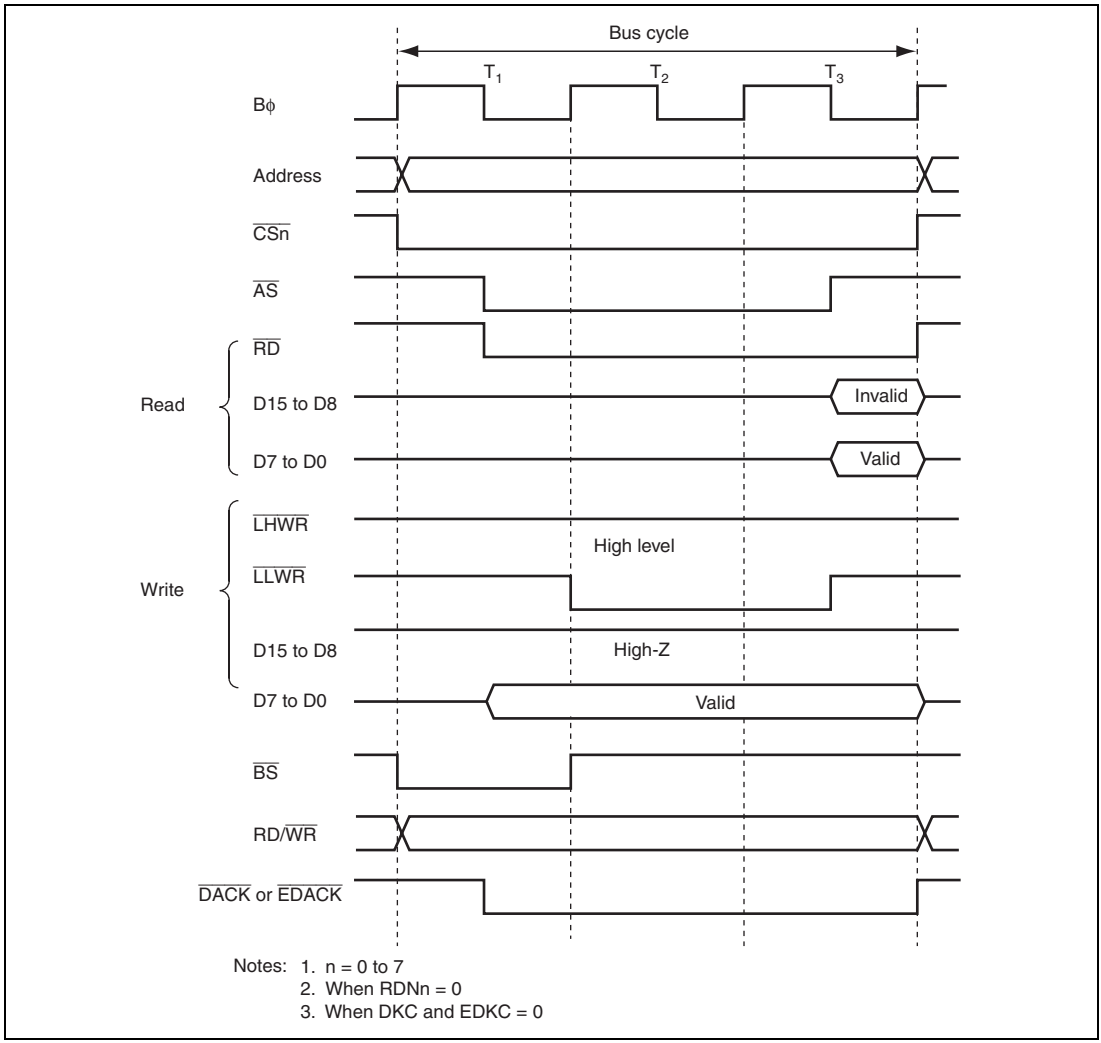



**Figure 9.19 16-Bit 3-State Access Space Bus Timing (Word Access for Odd Address)**

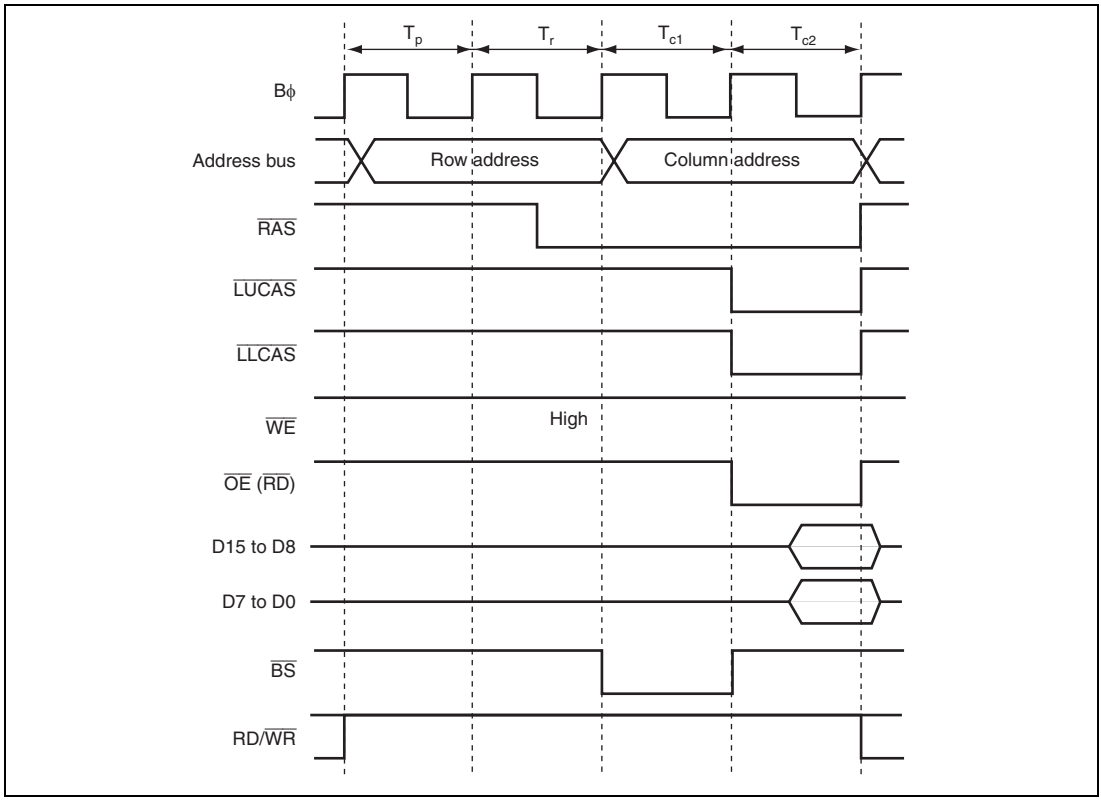



**Figure 9.46 Timing Example of Word Control with Use of Two CAS Signals (Read Access with Lowest Bit of Address = B'0, RAST = 0, CAST = 0)**

### 9.12.2 Pin States in Idle Cycle

Table 9.31 shows the pin states in an idle cycle.

**Table 9.31 Pin States in Idle Cycle**

| Pins | Pin State |
|---|---|
| A23 to A0 | Contents of following bus cycle |
| D15 to D0 | High impedance |
| $\overline{CSn}$ (n = 7 to 0) | High*1 |
| $\overline{LUCAS}$, $\overline{LLCAS}$ | High |
| DQMLU, DQMLL | High*2 |
| $\overline{AS}$ | High |
| $\overline{RD}$ | High |
| $\overline{BS}$ | High |
| RD/$\overline{WR}$ | High*3 |
| $\overline{AH}$ | low |
| $\overline{LHWR}$, $\overline{LLWR}$ | High |
| $\overline{LUB}$, $\overline{LLB}$ | High |
| CKE | High |
| $\overline{OE}$ | High |
| $\overline{RAS}$ | High/Low*4 |
| $\overline{CAS}$ | High |
| $\overline{WE}$ | High |
| $\overline{DACKn}$ (n = 3 to 0) | High |
| $\overline{EDACKn}$ (n = 3 to 0) | High |

Notes: 1. Low when accessing the SDRAM in full access cycle
2. Low when reading the SDRAM in full access cycle
3. Low when accessing or writing to the DRAM/SDRAM in full access cycle
4. The pin state varies depending on the DRAM space access/ area access other than the DRAM space, or RAS up mode/RAS down mode. For details, see figures 9.98 and 9.100.

**Table 10.2 Data Access Size, Valid Bits, and Settable Size**

| Mode | Data Access Size | BKSZH Valid Bits | BKSZ Valid Bits | Settable Size (Byte) |
|---|---|---|---|---|
| Repeat transfer and block transfer | Byte | 31 to 16 | 15 to 0 | 1 to 65,536 |
| | Word | | | 2 to 131,072 |
| | Longword | | | 4 to 262,144 |

### 10.3.6 DMA Mode Control Register (DMDR)

DMDR controls the DMAC operation.

- DMDR_0

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Bit Name | DTE | DACKE | TENDE | — | DREQS | NRD | — | — |
| 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 | R |
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit Name | ACT | — | — | — | ERRF | — | ESIF | DTIF |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/(W)* | R | R/(W)* | R/(W)* |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Bit Name | DTSZ1 | DTSZ0 | MDS1 | MDS0 | TSEIE | — | ESIE | DTIE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit Name | DTF1 | DTF0 | DTA | — | — | DMAP2 | DMAP1 | DMAP0 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R/W | R/W | R/W |

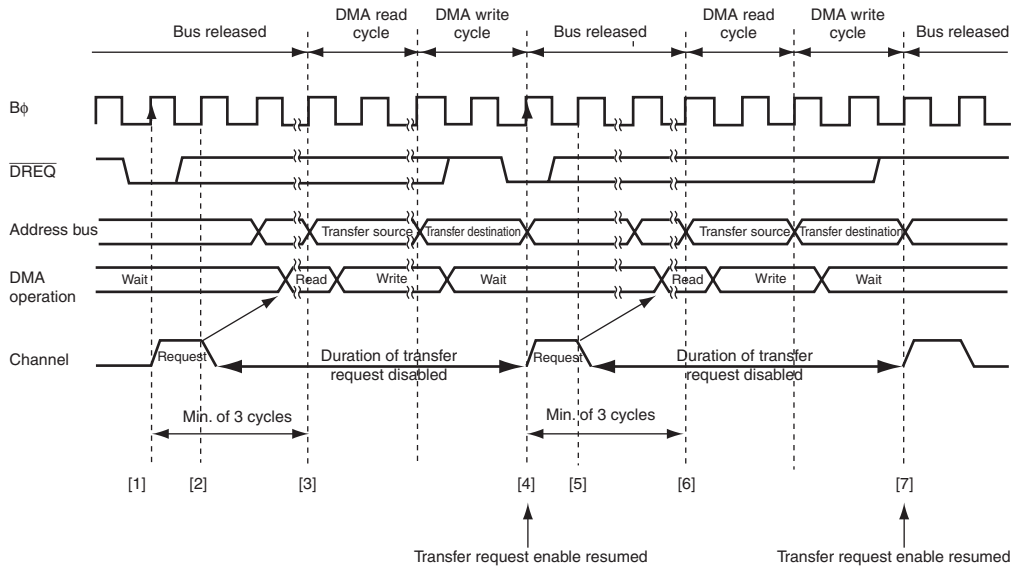
Note: * Only 0 can be written to this bit after having been read as 1, to clear the flag.

### (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**

### (6) ACT bit in EDMDR

The ACT bit in EDMDR indicates whether the EXDMAC is in standby or active state. When DTE = 0 and DTE = 1 (transfer request wait status) are specified, the ACT bit is set to 0. In another case (EXDMAC in the active state), the ACT bit is set to 1. The ACT bit is held to 1 during EXDMA transfer even if 0 is written to the DTE bit to halt transfer.

In block transfer mode, a block-size transfer is not halted even if 0 is written to the DTE bit to halt transfer. The ACT bit is held to 1 until a block-size transfer completes after 0 is written to the DTE bit.

In burst mode, transfer is halted after up to three times of EXDMA transfers are performed since the bus cycle in which 0 is written to the DTE bit has been processed. The ACT bit is held to 1 between termination of the last EXDMA cycle and 0-write in the DTE bit.

### (7) ERRF bit in EDMDR

This bit specifies termination of transfer by EXDMAC clearing the DTE bit to 0 for all channels if an address error or NMI interrupt is generated. The EXDMAC also sets 1 to the ERRF bit of EDMDR_0 regardless of the EXDMAC operation to indicate that an address error or NMI interrupt is generated. However, when an address error or an NMI interrupt has been generated in EXDMAC module stop mode, the ERRF bit is not set to 1.

### (8) ESIF bit in EDMDR

The ESIF bit in EDMDR is set to 1 when a transfer size interrupt, repeat size end interrupt, or an extended repeat area overflow interrupt is requested. When the ESIF bit is set to 1 and the ESIE bit in EDMDR is set to 1, a transfer escape interrupt is requested to the CPU or DTC.

The timing that the ESIF bit is set to 1 is when the EXDMA transfer bus cycle (the source of an interrupt request) terminates, the ACT bit in EDMDR is set to 0, and transfer is terminated.

When the DTE bit is set to 1 to resume transfer during interrupt processing, the ESIF bit is automatically cleared to 0 to cancel the interrupt request.

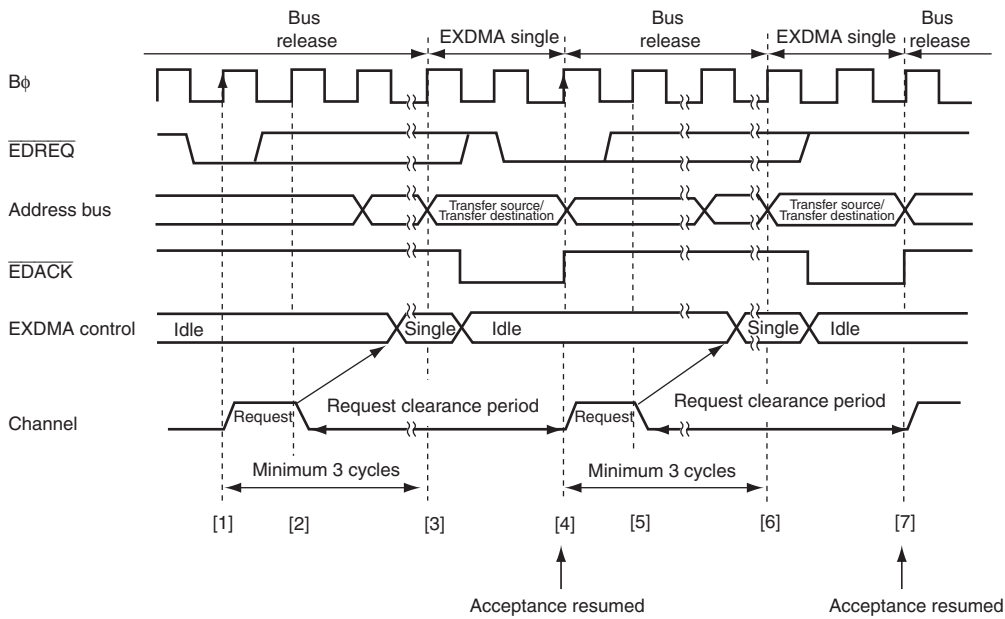
For details on interrupts, see section 11.9, Interrupt Sources.

### (3) $\overline{\text{EDREQ}}$ Pin Falling Edge Activation Timing

Figure 11.36 shows an example of single address mode transfer activated by the $\overline{\text{EDREQ}}$ pin falling edge.

$\overline{\text{EDREQ}}$ pin sampling is performed in each cycle starting at the next rise of Bϕ after the end of the DTE bit write cycle.

When a low level is sampled at the $\overline{\text{EDREQ}}$ pin while acceptance of a transfer request via the $\overline{\text{EDREQ}}$ pin is possible, the request is held within the EXDMAC. Then when activation is initiated within the EXDMAC, the request is cleared, and $\overline{\text{EDREQ}}$ pin high level sampling for edge sensing is started. If $\overline{\text{EDREQ}}$ pin high level sampling is completed by the end of the EXDMA single cycle, acceptance resumes after the end of the single cycle, and $\overline{\text{EDREQ}}$ pin low level sampling is performed again. This sequence of operations is repeated until the end of the transfer.




[1] Acceptance after transfer enabling; $\overline{\text{EDREQ}}$ pin low level is sampled at rise of Bϕ, and request is held.
[2], [5] Request is cleared at end of next bus cycle, and activation is started in EXDMAC.
[3], [6] EXDMA cycle starts; $\overline{\text{EDREQ}}$ pin high level sampling is started at rise of Bϕ.
[4], [7] When $\overline{\text{EDREQ}}$ pin high level has been sampled, acceptance is resumed after completion of write cycle.
(As in [1], $\overline{\text{EDREQ}}$ pin low level is sampled at rise of Bϕ, and request is held.)

**Figure 11.36 Example of Single Address Mode Transfer Activated by $\overline{\text{EDREQ}}$ Pin Falling Edge**

### (3) Transfer End by Repeat Size End Interrupt

In repeat transfer mode, when the RPTIE bit in EDACR is set to 1 and the next transfer request is generated on completion of a repeat-size transfer, a repeat size end interrupt request is generated. The interrupt request terminates EXDMA transfer, the DTE bit in EDMDR is cleared to 0, and the ESIF bit in EDMDR is set to 1 at the same time. If the DTE bit is set to 1 in this state, transfer resumes.

In block transfer or cluster transfer mode, a repeat size end interrupt request can be generated. In block transfer mode, if the next transfer request is generated at the end of a block-size transfer, a repeat size end interrupt request is generated. In cluster transfer mode, if the next transfer request is generated at the end of a cluster-size transfer, a repeat size end interrupt request is generated.

### (4) Transfer End by Extended Repeat Area Overflow Interrupt

If an address overflows the extended repeat area when an extended repeat area specification has been made and the SARIE or DARIE bit in EDACR is set to 1, an extended repeat area overflow interrupt is requested. The interrupt request terminates EXDMA transfer, the DTE bit in EDMDR is cleared to 0, and the ESIF bit in EDMDR is set to 1 at the same time.

In dual address mode, if an extended repeat area overflow interrupt is requested during a read cycle, the following write cycle processing is still executed.

In block transfer mode, if an extended repeat area overflow interrupt is requested during transfer of a block, transfer continues to the end of the block. Transfer end by means of an extended repeat area overflow interrupt occurs between block-size transfers.

In cluster transfer mode, if an extended repeat area overflow interrupt is requested during transfer of a cluster, transfer continues to the end of the cluster. Transfer end by means of an extended repeat area overflow interrupt occurs between cluster-size transfers.

### (5) Transfer End by 0-Write to DTE Bit in EDMDR

When 0 is written to the DTE bit in EDMDR by the CPU, etc., transfer ends after completion of the EXDMA cycle in which transfer is in progress or a transfer request was accepted.

In block transfer mode, EXDMA transfer ends after completion of one-block-size transfer in progress.

In cluster transfer mode, EXDMA transfer ends after completion of one-cluster-size transfer in progress.

### (7) PJ1/PO17/TIOCA6/TIOCB6

The pin function is switched as shown below according to the combination of register setting of PPG and TPU, setting of the port function control register (PFCR), and the PJ1DDR bit settings.

| | | Setting | | |
|---|---|---|---|---|
| | | PPG | TPU | I/O Port |
| Module Name | Pin Function | PO17_OE | TIOCB6_OE | PJ1DDR |
| PPG | PO17 output* | 1 | — | — |
| TPU | TIOCB6 output* | 0 | 1 | — |
| I/O port | PJ1 output* | 0 | 0 | 1 |
| | PJ1 input* | 0 | 0 | 0 |

Note: * Valid when PCJKE = 1.

### (8) PJ0/PO16/TIOCA6

The pin function is switched as shown below according to the combination of register setting of PPG and TPU, setting of the port function control register (PFCR), and the PJ0DDR bit settings.

| | | Setting | | |
|---|---|---|---|---|
| | | PPG | TPU | I/O Port |
| Module Name | Pin Function | PO16_OE | TIOCA6_OE | PJ0DDR |
| PPG | PO16 output* | 1 | — | — |
| TPU | TIOCA6 output* | 0 | 1 | — |
| I/O port | PJ0 output* | 0 | 0 | 1 |
| | PJ0 input* | 0 | 0 | 0 |

Note: * Valid when PCJKE = 1.

**Table 14.22 TIORH_0**

| | | | | Description | |
|---|---|---|---|---|---|
| **Bit 3 IOA3** | **Bit 2 IOA2** | **Bit 1 IOA1** | **Bit 0 IOA0** | **TGRA_0 Function** | **TIOCA0 Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output disabled |
| 0 | 0 | 0 | 1 | | Initial output is 0 output<br>0 output at compare match |
| 0 | 0 | 1 | 0 | | Initial output is 0 output<br>1 output at compare match |
| 0 | 0 | 1 | 1 | | Initial output is 0 output<br>Toggle output at compare match |
| 0 | 1 | 0 | 0 | | Output disabled |
| 0 | 1 | 0 | 1 | | Initial output is 1 output<br>0 output at compare match |
| 0 | 1 | 1 | 0 | | Initial output is 1 output<br>1 output at compare match |
| 0 | 1 | 1 | 1 | | Initial output is 1 output<br>Toggle output at compare match |
| 1 | 0 | 0 | 0 | Input capture register | Capture input source is TIOCA0 pin<br>Input capture at rising edge |
| 1 | 0 | 0 | 1 | | Capture input source is TIOCA0 pin<br>Input capture at falling edge |
| 1 | 0 | 1 | x | | Capture input source is TIOCA0 pin<br>Input capture at both edges |
| 1 | 1 | x | x | | Capture input source is channel 1/count clock<br>Input capture at TCNT_1 count-up/count-down* |

[Legend]

x: Don't care

Note: * When the bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and Pϕ/1 is used as the count clock of TCNT_1, this setting is invalid and input capture is not generated.

## 14.4 Operation

### 14.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

#### (1) Counter Operation

When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

#### (a) Example of count operation setting procedure

Figure 14.3 shows an example of the count operation setting procedure.

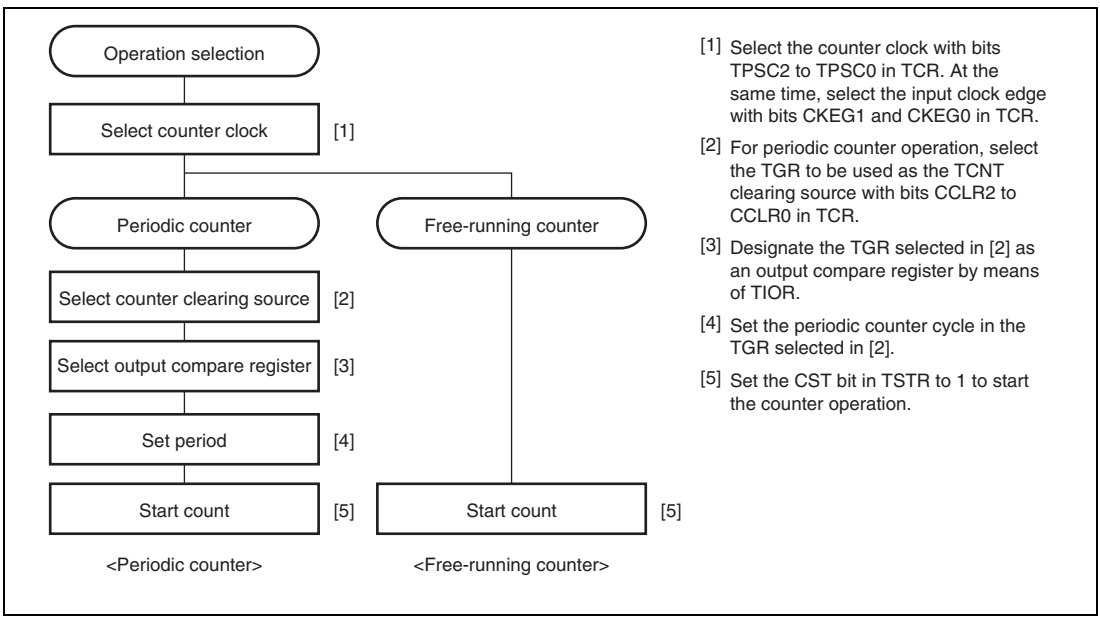
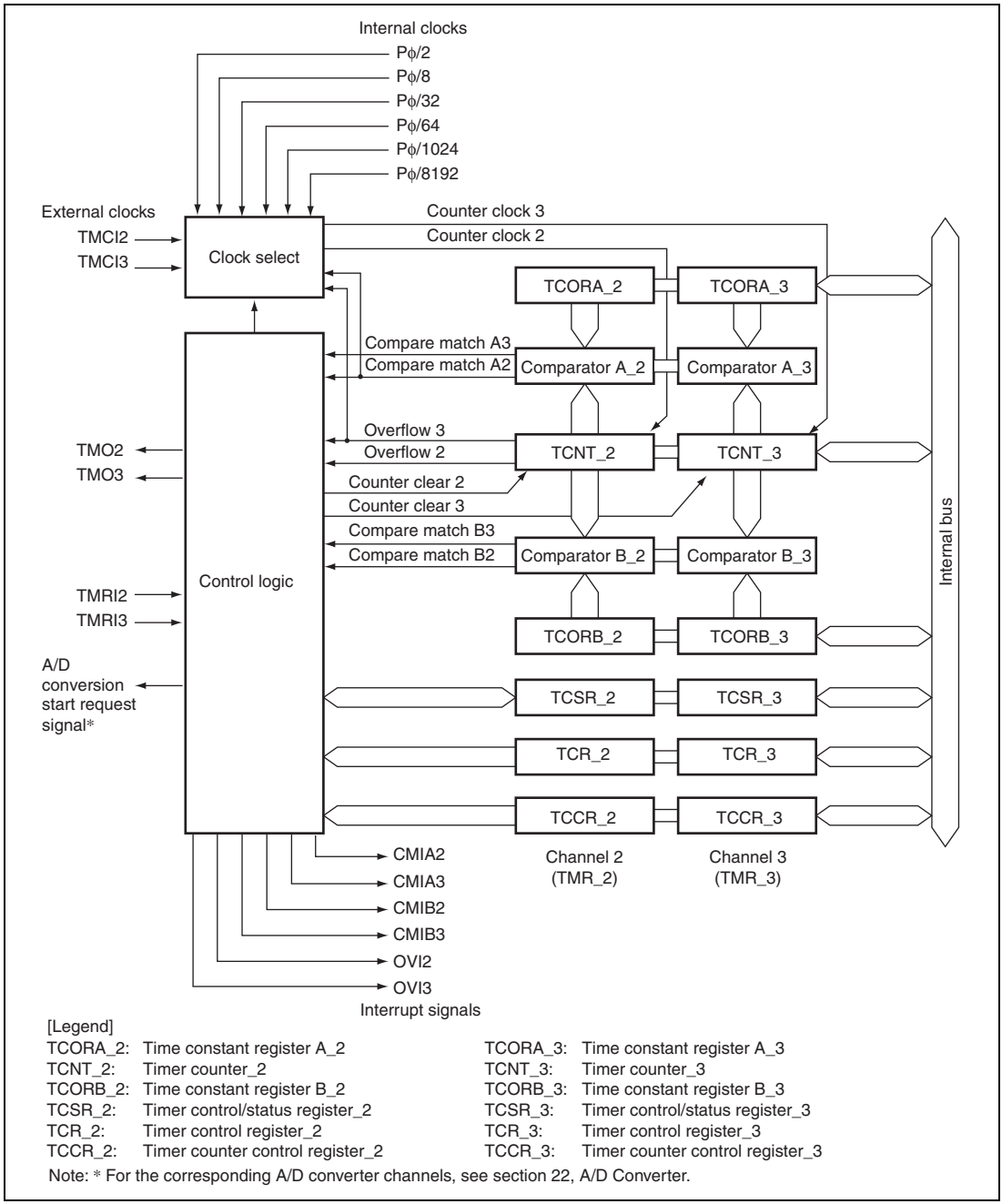



**Figure 14.3 Example of Counter Operation Setting Procedure**




[Legend]

| | | | |
|---|---|---|---|
| TCORA_2: | Time constant register A_2 | TCORA_3: | Time constant register A_3 |
| TCNT_2: | Timer counter_2 | TCNT_3: | Timer counter_3 |
| TCORB_2: | Time constant register B_2 | TCORB_3: | Time constant register B_3 |
| TCSR_2: | Timer control/status register_2 | TCSR_3: | Timer control/status register_3 |
| TCR_2: | Timer control register_2 | TCR_3: | Timer control register_3 |
| TCCR_2: | Timer counter control register_2 | TCCR_3: | Timer counter control register_3 |

Note: * For the corresponding A/D converter channels, see section 22, A/D Converter.

**Figure 16.2 Block Diagram of 8-Bit Timer Module (Unit 1)**

**Table 16.4 Clock Input to TCNT and Count Condition (Unit 2)**

| | TCR | | | TCCR | | |
|---|---|---|---|---|---|---|
| Channel | Bit 2 CKS2 | Bit 1 CKS1 | Bit 0 CKS0 | Bit 1 ICKS1 | Bit 0 ICKS0 | Description |
| TMR_4 | 0 | 0 | 0 | — | — | Clock input prohibited |
| | 0 | 0 | 1 | 0 | 0 | Uses internal clock. Counts at rising edge of Pϕ/8. |
| | | | | 0 | 1 | Uses internal clock. Counts at rising edge of Pϕ/2. |
| | | | | 1 | 0 | Uses internal clock. Counts at falling edge of Pϕ/8. |
| | | | | 1 | 1 | Uses internal clock. Counts at falling edge of Pϕ/2. |
| | 0 | 1 | 0 | 0 | 0 | Uses internal clock. Counts at rising edge of Pϕ/64. |
| | | | | 0 | 1 | Uses internal clock. Counts at rising edge of Pϕ/32. |
| | | | | 1 | 0 | Uses internal clock. Counts at falling edge of Pϕ/64. |
| | | | | 1 | 1 | Uses internal clock. Counts at falling edge of Pϕ/32. |
| | 0 | 1 | 1 | 0 | 0 | Uses internal clock. Counts at rising edge of Pϕ/8192. |
| | | | | 0 | 1 | Uses internal clock. Counts at rising edge of Pϕ/1024. |
| | | | | 1 | 0 | Uses internal clock. Counts at rising edge of Pϕ. |
| | | | | 1 | 1 | Uses internal clock. Counts at falling edge of Pϕ/1024. |
| | 1 | 0 | 0 | — | — | Counts at TCNT_5 overflow signal*. |
| TMR_5 | 0 | 0 | 0 | — | — | Clock input prohibited |
| | 0 | 0 | 1 | 0 | 0 | Uses internal clock. Counts at rising edge of Pϕ/8. |
| | | | | 0 | 1 | Uses internal clock. Counts at rising edge of Pϕ/2. |
| | | | | 1 | 0 | Uses internal clock. Counts at falling edge of Pϕ/8. |
| | | | | 1 | 1 | Uses internal clock. Counts at falling edge of Pϕ/2. |
| | 0 | 1 | 0 | 0 | 0 | Uses internal clock. Counts at rising edge of Pϕ/64. |
| | | | | 0 | 1 | Uses internal clock. Counts at rising edge of Pϕ/32. |
| | | | | 1 | 0 | Uses internal clock. Counts at falling edge of Pϕ/64. |
| | | | | 1 | 1 | Uses internal clock. Counts at falling edge of Pϕ/32. |
| | 0 | 1 | 1 | 0 | 0 | Uses internal clock. Counts at rising edge of Pϕ/8192. |
| | | | | 0 | 1 | Uses internal clock. Counts at rising edge of Pϕ/1024. |
| | | | | 1 | 0 | Uses internal clock. Counts at rising edge of Pϕ. |
| | | | | 1 | 1 | Uses internal clock. Counts at falling edge of Pϕ/1024. |
| | 1 | 0 | 0 | — | — | Counts at TCNT_4 compare match A*. |
| All | 1 | 0 | 1 | — | — | Setting prohibited |
| | 1 | 1 | 0 | — | — | Setting prohibited |
| | 1 | 1 | 1 | — | — | Setting prohibited |

Note: * If the clock input of channel 4 is the TCNT_5 overflow signal and that of channel 5 is the TCNT_4 compare match signal, no incrementing clock is generated. Do not use this setting.

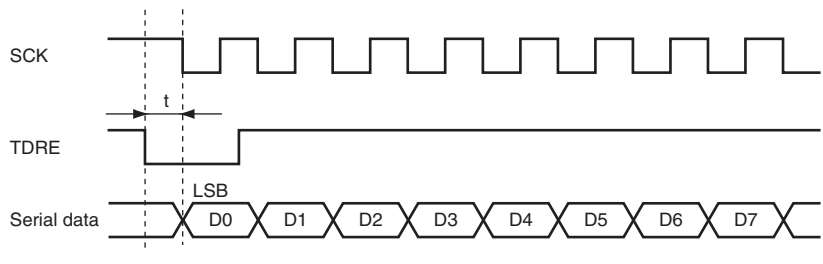
### 19.10.5 Relation between Writing to TDR and TDRE Flag

The TDRE flag in SSR is a status flag which indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR irrespective of the TDRE flag status. However, if new data is written to TDR when the TDRE flag is 0, that is, when the previous data has not been transferred to TSR yet, the previous data in TDR is lost. Be sure to write transmit data to TDR after verifying that the TDRE flag is set to 1.

### 19.10.6 Restrictions on Using DTC or DMAC

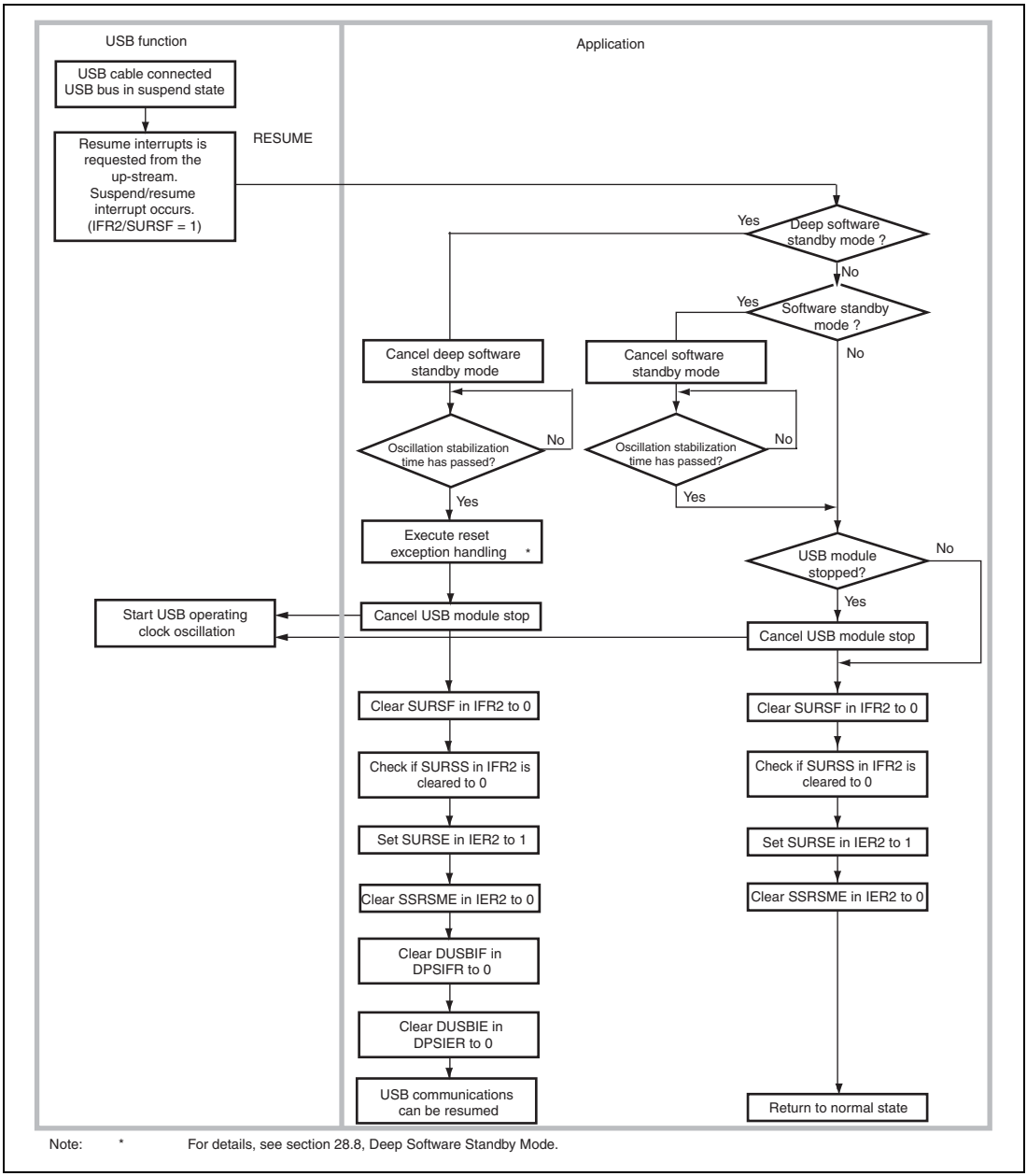
- When the external clock source is used as a synchronization clock, update TDR by the DMAC or DTC and wait for at least five Pϕ clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR modification, the SCI may malfunction (see figure 19.38).
- When using the DMAC or DTC to read RDR, be sure to set the receive end interrupt (RXI) as the DTC or DMAC activation source.




Note: When external clock is supplied, t must be more than four clock cycles.

**Figure 19.38 Sample Transmission using DTC in Clocked Synchronous Mode**

- The DTC is not activated by the RXI or TXI request by SCI_5 or SCI6.




Note: * For details, see section 28.8, Deep Software Standby Mode.

**Figure 20.5 Resume Operation from Up-Stream**

### 28.9.4 Timing Sequence at Power-On

Figure 28.9 shows the timing sequence at power-on.

At power-on, the $\overline{\text{RES}}$ pin must be driven low with the $\overline{\text{STBY}}$ pin driven high for a given time in order to clear the reset state.

To enter hardware standby mode immediately after power-on, drive the $\overline{\text{STBY}}$ pin low after exiting the reset state.

For details on clearing hardware standby mode, see section 28.9.3, Hardware Standby Mode Timing.

In a power-on reset*, power on while driving the $\overline{\text{STBY}}$ or $\overline{\text{RES}}$ pin to a high-level.

Note: * Supported only by the H8SX/1668M Group.

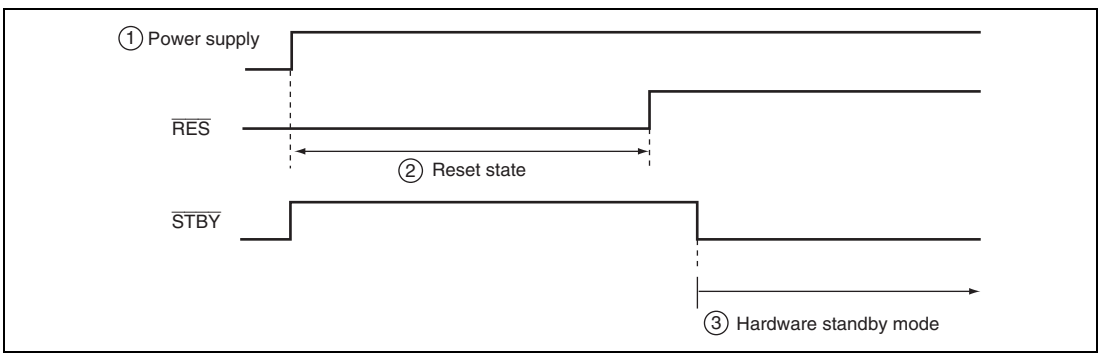



**Figure 28.9 Timing Sequence at Power-On**

## 30.3 DC Characteristics H8SX/1668M Group

### Table 30.4 DC Characteristics (1)

Conditions: $V_{CC}$ = $PLLV_{CC}$ = $DrV_{CC}$ = 2.95 V to 3.6 V, $AV_{CC}$ = 3.0 V to 3.6 V, $V_{ref}$ = 3.0 V to $AV_{CC}$, $V_{SS}$ = $PLLV_{SS}$ = $DrV_{SS}$ = $AV_{SS}$ = 0 V*[1], $T_a$ = –20°C to +75°C (regular specifications), $T_a$ = –40°C to +85°C (wide-range specifications)

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Schmitt trigger input voltage | $\overline{IRQ}$ input pin, TPU input pin, TMR input pin, port 2, port 3, port J, port K | $VT^-$ | $V_{CC} \times 0.2$ | — | — | V | |
| | | $VT^+$ | — | — | $V_{CC} \times 0.7$ | V | |
| | | $VT^+ - VT^-$ | $V_{CC} \times 0.06$ | — | — | V | |
| | $\overline{IRQ0}$-B to $\overline{IRQ7}$-B input pins | $VT^-$ | $AV_{CC} \times 0.2$ | — | — | V | |
| | | $VT^+$ | — | — | $AV_{CC} \times 0.7$ | V | |
| | | $VT^+ - VT^-$ | $AV_{CC} \times 0.06$ | — | — | V | |
| Input high voltage (except Schmitt trigger input pin) | MD, $\overline{RES}$, $\overline{STBY}$, EMLE, NMI | $V_{IH}$ | $V_{CC} \times 0.9$ | — | $V_{CC} + 0.3$ | V | |
| | EXTAL | | $V_{CC} \times 0.7$ | — | $V_{CC} + 0.3$ | | |
| | Other input pins | | | | | | |
| | Port 5 | | $AV_{CC} \times 0.7$ | — | $AV_{CC} + 0.3$ | | |
| Input low voltage (except Schmitt trigger input pin) | MD, $\overline{RES}$, $\overline{STBY}$, EMLE | $V_{IL}$ | –0.3 | — | $V_{CC} \times 0.1$ | V | |
| | EXTAL, NMI | | –0.3 | — | $V_{CC} \times 0.2$ | | |
| | Other input pins | | –0.3 | — | $V_{CC} \times 0.2$ | | |
| Output high voltage | All output pins | $V_{OH}$ | $V_{CC}$ – 0.5 | — | — | V | $I_{OH}$ = –200 µA |
| | | | $V_{CC}$ – 1.0 | — | — | | $I_{OH}$ = –1 mA |
| Output low voltage | All output pins | $V_{OL}$ | — | — | 0.4 | V | $I_{OL}$ = 1.6 mA |
| | Port 3 | | — | — | 1.0 | | $I_{OL}$ = 10 mA |
| Input leakage current | $\overline{RES}$ | $\lvert I_{in} \rvert$ | — | — | 10.0 | µA | $V_{in}$ = 0.5 to $V_{CC}$ – 0.5 V |
| | MD, $\overline{STBY}$, EMLE, NMI | | — | — | 1.0 | | |
| | Port 5 | | — | — | 1.0 | | $V_{in}$ = 0.5 to $AV_{CC}$ – 0.5 V |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 28.8.2 Exit from Deep Software Standby Mode | 1267 | Replaced<br>Overview<br>1. Exit from software standby mode by interrupt<br>Replaced due to the addition of the LVD. |
| | | Added<br>2. Exit from voltage monitoring reset*2<br>3. Exit from power-on reset*2<br>Added due to the addition of the voltage monitoring reset and power-on reset. |
| 28.9.4 Timing Sequence at Power-On | 1280 | Replaced<br>Replaced due to the addition of the power-on reset. |
| 28.12.7 Conflict between a transition to deep software standby mode and interrupts | 1286 | Replaced<br>Replaced due to the addition of the voltage monitoring interrupt. |
| Section 29 List of Registers<br>29.1 Register Addresses (Address Order) | 1300 | Added<br>Added due to the addition of the LVD. |

| Register Name | Abbreviation | Number of Bits | Address | Module | Data Width | Access Cycles (Read/Write) |
|---|---|---|---|---|---|---|
| Low voltage detection control register*2 | LVDCR | 8 | H'FFE78 | SYSTEM | 8 | 2Iϕ/3Iϕ |