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	SH-2
Core Size	32-Bit Single-Core
Speed	16MHz
Connectivity	EBI/EMI, SCI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	74
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	112-BQFP
Supplier Device Package	112-QFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/hd6417040avcf16v

Table 1.3 Pin Arrangement by Mode for SH7040, SH7042 (TQFP-120 Pin)

TQFP120 Pin No.	MCU Mode	PROM Mode
1	NC	NC
2	PE14/TIOC4C/DACK0/$\overline{AH}$	V_{CC}
3	PE15/TIOC4D/DACK1/$\overline{IRQOUT}$	CE
4	V_{SS}	V_{SS}
5	PC0/A0	A0
6	PC1/A1	A1
7	PC2/A2	A2
8	PC3/A3	A3
9	PC4/A4	A4
10	PC5/A5	A5
11	PC6/A6	A6
12	PC7/A7	A7
13	PC8/A8	A8
14	PC9/A9	NC
15	PC10/A10	A10
16	PC11/A11	A11
17	PC12/A12	A12
18	PC13/A13	A13
19	PC14/A14	A14
20	PC15/A15	A15
21	PB0/A16	A16
22	V_{CC}	V_{CC}
23	PB1/A17	NC
24	V_{SS}	V_{SS}
25	PB2/$\overline{IRQ0}$/$\overline{POE0}$/$\overline{RAS}$	NC
26	PB3/$\overline{IRQ1}$/$\overline{POE1}$/$\overline{CASL}$	OE
27	PB4/$\overline{IRQ2}$/$\overline{POE2}$/$\overline{CASH}$	PGM
28	V_{SS}	V_{SS}
29	PB5/$\overline{IRQ3}$/$\overline{POE3}$/RDWR	V_{CC}
30	NC	NC
31	NC	NC

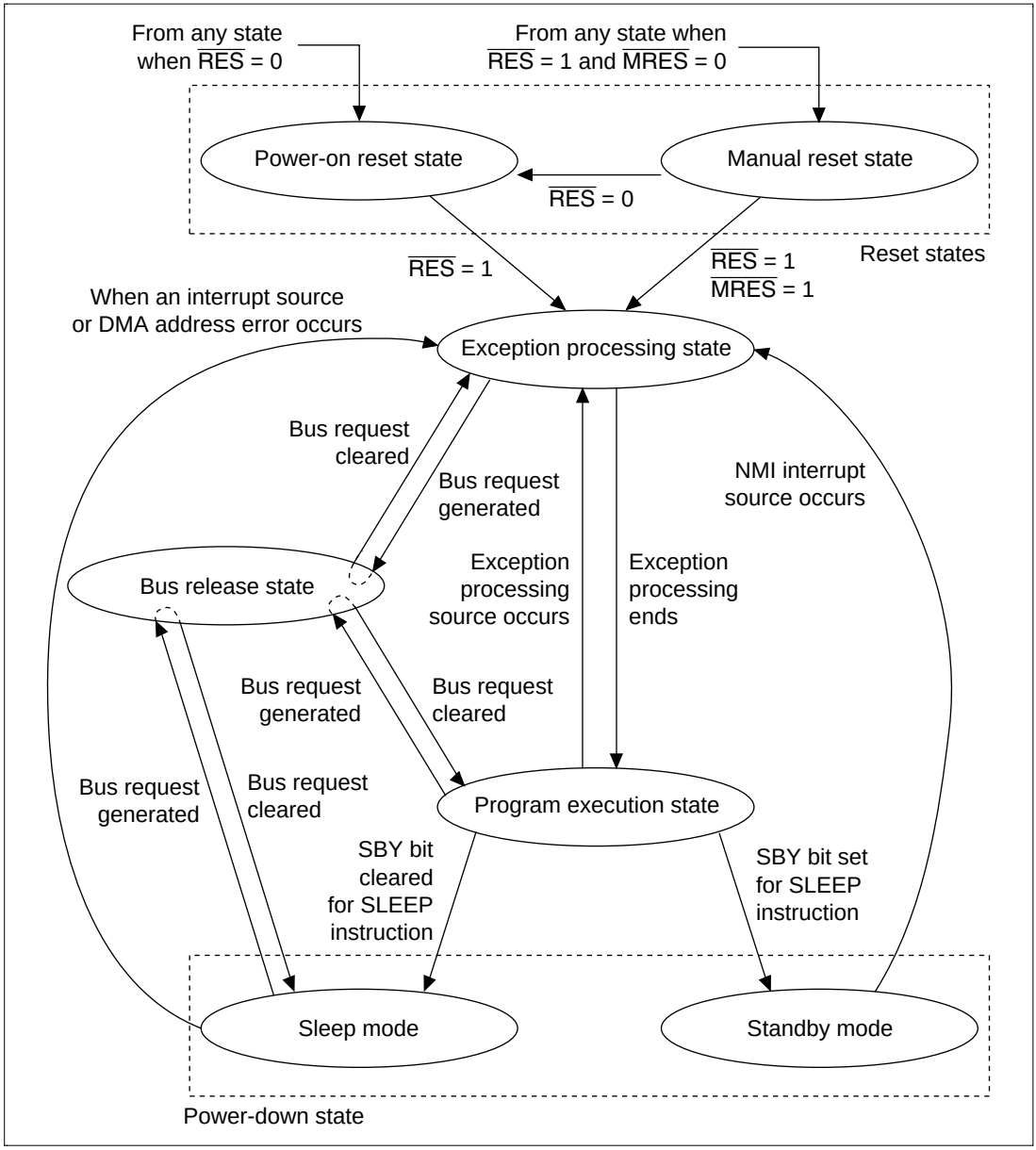

Figure 2.6 Transitions between Processing States

Reset State: The CPU resets in the reset state. When the $\overline{\text{RES}}$ pin level goes low, a power-on reset results. When the $\overline{\text{RES}}$ pin is high and MRES is low, a manual reset will occur.

Exception Processing State: The exception processing state is a transient state that occurs when exception processing sources such as resets or interrupts alter the CPU's processing state flow.

For the A mask, overwrite this register as follows:

When clearing bit to 0: read the 1 bit to clear and write 0.
When setting bit to 1: read the 0 bit to set and write 1.

Bit:	7	6	5	4	3	2	1	0
	DTE7	DTE6	DTE5	DTE4	DTE3	DTE2	DTE1	DTE0
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*

Note: * DTER bits can only be modified by writing 1 after reading 0, or writing 0 after reading 1.

8.2.8 DTC Control/Status Register (DTCSR)

The DTCSR is a 16-bit readable/writable register that sets disable/enable for DTC activation by software, as well as the DTC vector addresses for software activation. It also indicates the DTC transfer status.

The DTCSR is initialized to H'0000 by power-on resets and in standby mode. Manual reset does not initialize DTCSR.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	NMIF	AE	SWDTE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W*1	R/W*1	R/W*2

Bit:	7	6	5	4	3	2	1	0
Bit name:	DTVEC7	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*3	R/W*3	R/W*3	R/W*3	R/W*3	R/W*3	R/W*3	R/W*3 *4

Notes: *1 For the NMIF and AE bits, only a 0 write after a 1 read is possible.
*2 For the SWDTE bit, a 1 write is always possible, but a 0 write is possible only after a 1 is read.
*3 For the DTVEC7–DTVEC0 bits, writes are possible only when SWDTE = 0.
*4 Be sure to write 0 to the DTVEC0 bit.

Bits 15–11—Reserved: These bits always read as 0. The write value should always be 0.

Input Capture Operation: Figure 12.14 shows input capture. The falling edge of TIOCB and both edges of TIOCA are selected as input capture input edges. In the example, TCNT is set to clear at the input capture of the TGRB register.

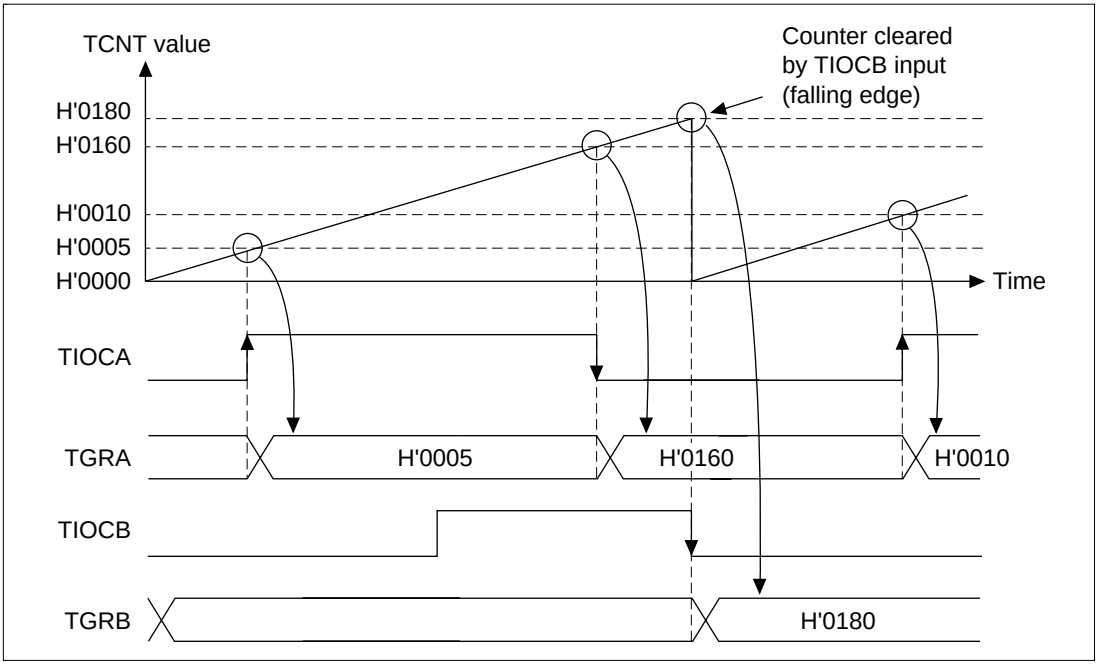

Figure 12.14 Input Capture Operation

12.4.3 Synchronous Operation

In the synchronizing mode, two or more timer counters can be rewritten simultaneously (synchronized preset). Multiple timer counters can also be cleared simultaneously using TCR settings (synchronized clear).

The synchronizing mode can increase the number of TGR registers for a single time base. All five channels can be set for synchronous operation.

12.6 Operation Timing

12.6.1 Input/Output Timing

TCNT Count Timing: Count timing for the TCNT counter with internal clock operation is shown in figure 12.59. Count timing with external clock operation (normal mode) is shown in figure 12.60, and figure 12.61 shows count timing with external clock operation (phase counting mode).

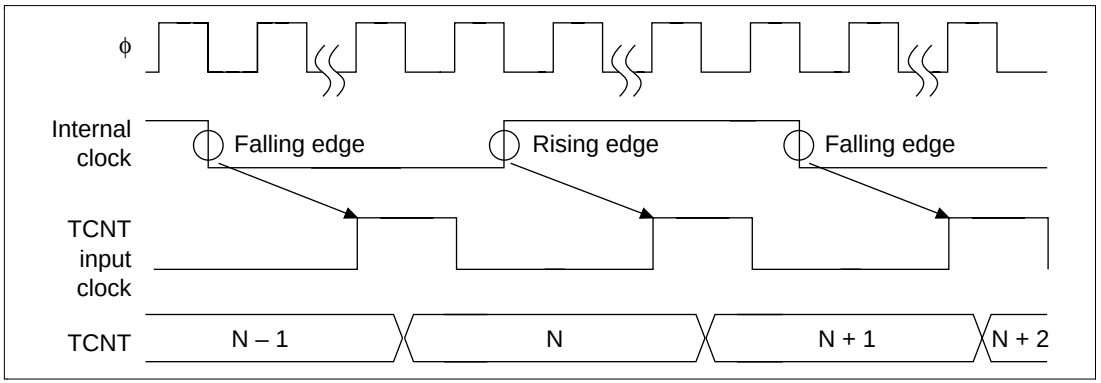

Figure 12.59 TCNT Count Timing during Internal Clock Operation

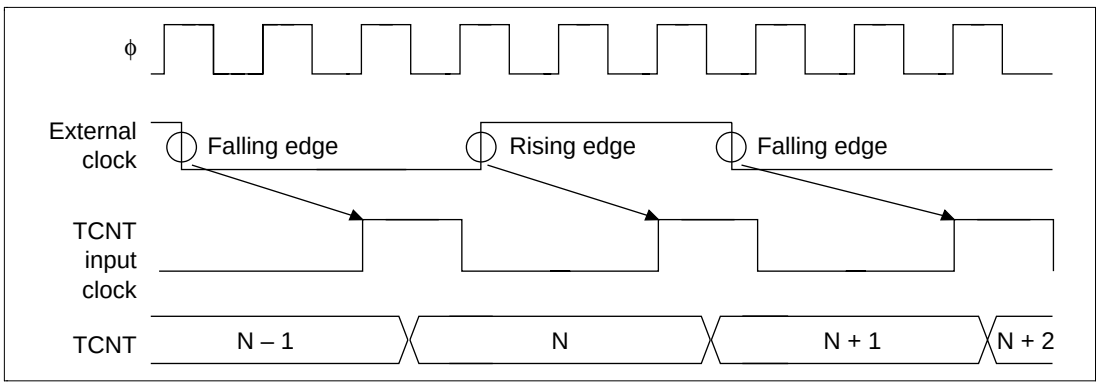

Figure 12.60 TCNT Count Timing during External Clock Operation (Normal Mode)

- In PWM mode 2, since a waveform is not output to the cycle register pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode and perform initialization with TIOR, then restore TIOR to its initial value, and temporarily disable channel 3 and 4 output with the timer output master enable register (TOER). Then operate the unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER setting).

Pin initialization procedures are described below for the numbered combinations in table 12.19. The active level is assumed to be low.

Note: Channel number is substituted for * indicated in this article.

12.10 POE Register Descriptions

12.10.1 Input Level Control/Status Register (ICSR)

The input level control/status register (ICSR) is a 16-bit read/write register that selects the $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin input modes, controls the enable/prohibit of interrupts, and indicates status. If any of the POE3F–POE0F bits are set to 1, the high current pins become high impedance state.

ICSR is initialized to H'0000 by power-on resets; however, it is not initialized for manual resets, standby mode, or sleep mode, so the previous data is maintained.

Bit:	15	14	13	12	11	10	9	8
	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	POE3M1	POE3M0	POE2M1	POE2M0	POE1M1	POE1M0	POE0M1	POE0M0
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

Note: * Only 0 writes are possible to clear the flags.

- Bit 15—POE3 Flag (POE3F): This flag indicates that a high impedance request has been input to the $\overline{\text{POE3}}$ pin.

Bit 15: POE3F	Description
0	Clear condition: By writing 0 to POE3F after reading a POE3F = 1 (initial value)
1	Set condition: When the input set by ICSR bits 7 and 6 occurs at the $\overline{\text{POE3}}$ pin

- Bit 14—POE2 Flag (POE2F): This flag indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin.

Bit 14: POE2F	Description
0	Clear condition: By writing 0 to POE2F after reading a POE2F = 1 (initial value)
1	Set condition: When the input set by ICSR bits 5 and 4 occurs at the $\overline{\text{POE2}}$ pin

13.2.4 Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in that they are more difficult to write to. The procedures for writing and reading these registers are given below.

Writing to the TCNT and TCSR: These registers must be written by a word transfer instruction. They cannot be written by byte transfer instructions.

The TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must be H'5A (for the TCNT) or H'A5 (for the TCSR) (figure 13.2). This transfers the write data from the lower byte to the TCNT or TCSR.

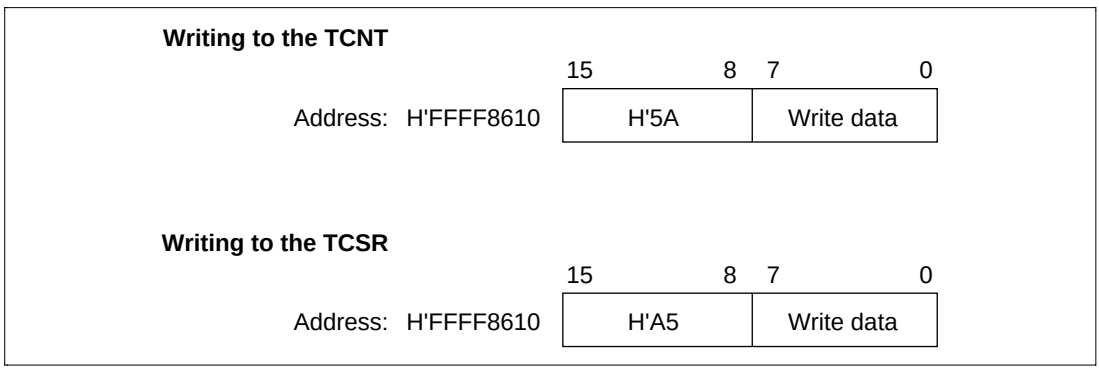

Figure 13.2 Writing to the TCNT and TCSR

Writing to the RSTCSR: The RSTCSR must be written by a word access to address H'FFFF8612. It cannot be written by byte transfer instructions.

Procedures for writing 0 in WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 13.3.

To write 0 in the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

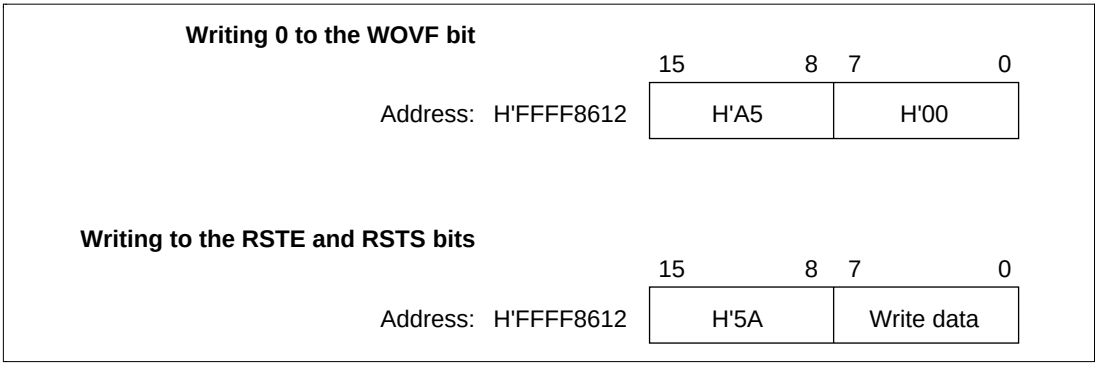

Figure 13.3 Writing to the RSTCSR

Reading from the TCNT, TCSR, and RSTCSR: TCNT, TCSR, and RSTCSR are read like other registers. Use byte transfer instructions. The read addresses are H'FFFF8610 for the TCSR, H'FFFF8611 for the TCNT, and H'FFFF8613 for the RSTCSR.

13.3 Operation

13.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits of the TCSR to 1. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. No TCNT overflows will occur while the system is operating normally, but if the TCNT fails to be rewritten and overflows occur due to a system crash or the like, a $\overline{WDTOVF}$ signal is output externally (figure 13.4). The $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 128 ϕ clock cycles.

If the RSTE bit in the RSTCSR is set to 1, a signal to reset the chip will be generated internally simultaneous to the $\overline{WDTOVF}$ signal when TCNT overflows. Either a power-on reset or a manual reset can be selected by the RSTS bit. The internal reset signal is output for 512 ϕ clock cycles.

When a watchdog overflow reset is generated simultaneously with a reset input at the $\overline{RES}$ pin, the $\overline{RES}$ reset takes priority, and the WOVF bit is cleared to 0.

The following are not initialized a WDT reset signal:

- The MTU’s POE (Port Output Enable) function register
- PFC (Pin Function Controller) function register
- I/O port register

Initializing is only possible by external power-on reset.

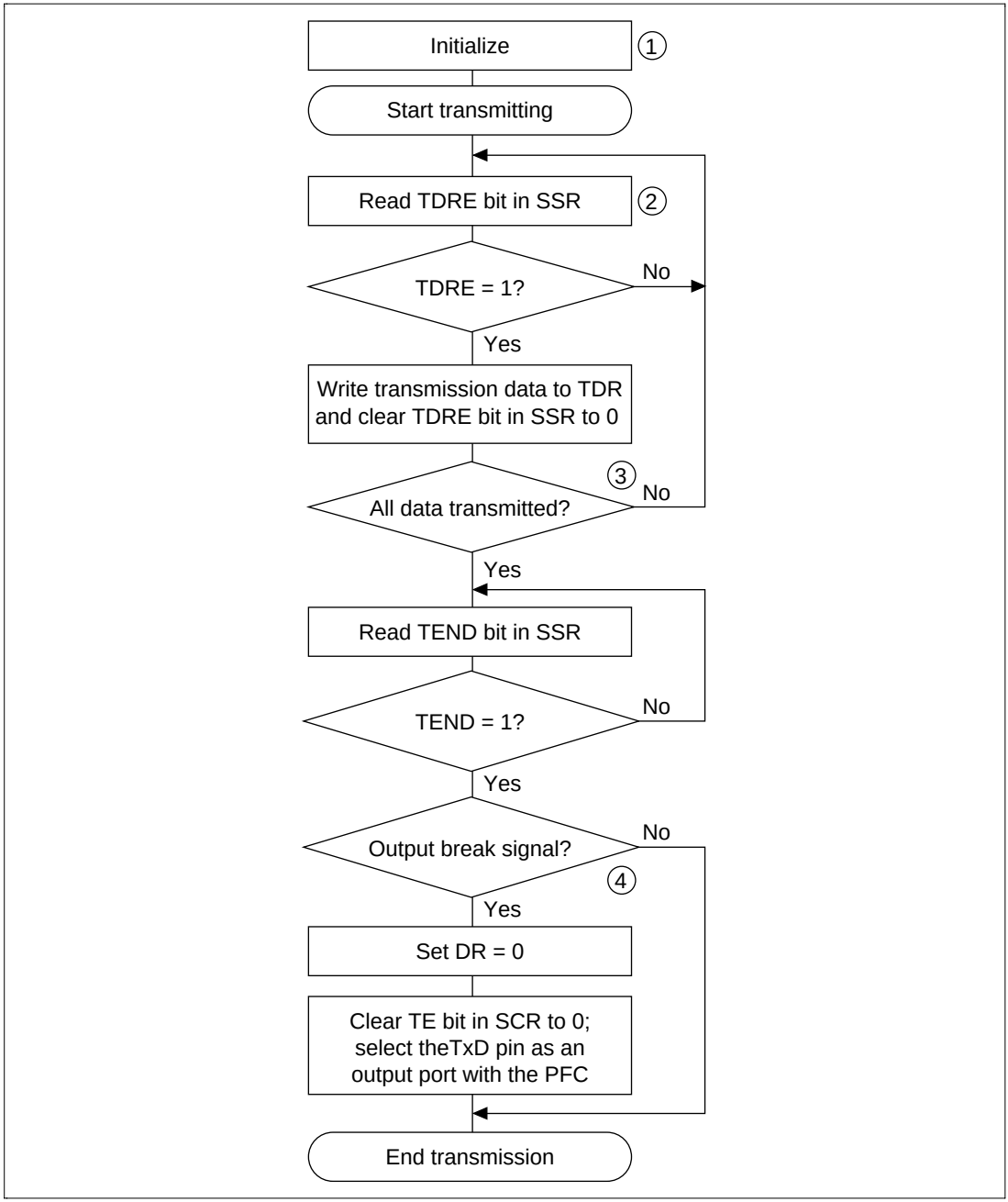

Figure 14.5 Sample Flowchart for Transmitting Serial Data

Section 15 High Speed A/D Converter (Excluding A Mask)

15.1 Overview

The high speed A/D converter has 10-bit resolution, and can select from a maximum of eight channels of analog inputs.

15.1.1 Features

The high speed A/D converter has the following features:

- 10-bit resolution
- Eight input channels
- Analog conversion voltage range setting is selectable
  - Using the reference voltage pin (AVref) as an analog standard voltage (Vref), conversion of analog input from 0 to Vref (only with SH7043).
- High-speed conversion
  - Minimum conversion time: 2.9 µs per channel (for 28-MHz operation)
  - 1.4 µs per channel during continuous conversion
- Multiple conversion modes
  - Select mode/group mode
  - Single mode/scan mode
  - Buffered operation possible
  - 2 channel simultaneous sampling possible
- Three types of conversion start
  - Software, timer conversion start trigger (MTU), or $\overline{\text{ADTRG}}$ pin can be selected.
- Eight data registers
  - Conversion results stored in 16-bit data registers corresponding to each channel.
- Sample and hold function
- A/D conversion end interrupt generation
  - An A/D conversion end interrupt (ADI) request can be generated on completion of A/D conversions

- Bit 2—Simultaneous Sampling (DSMP): Enables or disables the simultaneous sampling of two channels. See section 15.4.6, Simultaneous Sampling Operation, for details on simultaneous sampling.

  Set the DSMP bit only while conversion is halted.

Bit 2: DSMP	Description
0	Normal sampling operation (initial value)
1	Simultaneous sampling operation

- Bits 1–0—Buffer Enable 1, 0 (BUFE1, BUFE0): These bits select whether to use the ADDRB–ADDRD as buffer registers.

  Set the BUFE1 and BUFE0 bits only while conversion is halted.

Bit 1: BUFE1	Bit 0: BUFE0	Description
0	0	Normal operation (initial value)
0	1	ADDRA and ADDRB buffer operation: conversion result → ADDRA → ADDRB (ADDRB is the buffer register)
1	0	ADDRA and ADDRC, also ADDRB and ADDRD buffer operation: conversion result 1 → ADDRA → ADDRC, conversion result 2 → ADDRB → ADDRD (ADDRC and ADDRD are buffer registers)
1	1	ADDRA–ADDRD buffer operation: conversion result → ADDRA → ADDRB → ADDRC → ADDRD (ADDRB–ADDRD are buffer registers)

15.3 Bus Master Interface

The ADDRA–ADDRH are 16-bit registers with a 16-bit width data bus to the bus master. The bus master can read from ADDRA–ADDRH in either word or byte units.

When an ADDR is read in word units, the ADDR contents are transferred to the bus master 16 bits at a time. In byte unit reads, the contents of the most significant eight bits (AD9–AD2) of the converted data (AD9–AD0) are transferred to the bus master.

Figures 15.2 and 15.3 shows an example of the ADDR read operation.

16.2 Register Descriptions

16.2.1 A/D Data Register A–D (ADDRA0–ADDRD0, ADDRA1–ADDRD1)

A/D registers are special registers that read stored results of A/D conversion in 16 bits. There are eight registers: ADDRA0–ADDRD0 (A/D0) and ADDRA1–ADDRD1 (A/D1).

The A/D converted data is 10 bit data which is to the ADDR of the corresponding converted channel for storage. The upper 8 bits of the A/D converted data correspond to the upper byte of the ADDR and the lower 2 bits correspond to the lower byte. Bits 5–0 of the lower byte of ADDR are reserved and always read 0. Analog input channels and correspondence to ADDR are shown in table 16.3.

ADDR can always be read from the CPU. The upper byte may be read directly. The lower byte is transferred through the temporary register (TEMP). For details, see section 16.3, Interface with CPU.

ADDR is initialized to H'0000 during power-on reset or standby mode. ADDR will not be initialized by manual reset.

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRn :	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

(n=A to D)

Table 16.3 Analog Input Channel and ADDRA–ADDRD Correspondence

Analog Input Channel	A/D Data Register	Module
AN0	ADDRA0	A/D0
AN1	ADDRB0	
AN2	ADDRC0	
AN3	ADDRD0	
AN4	ADDRA1	A/D1
AN5	ADDRB1	
AN6	ADDRC1	
AN7	ADDRD1	

Section 17 Compare Match Timer (CMT)

17.1 Overview

The SH7040 series has an on-chip compare match timer (CMT) configured of 16-bit timers for two channels. The CMT has 16-bit counters and can generate interrupts at set intervals.

17.1.1 Features

The CMT has the following features:

- Four types of counter input clock can be selected
  - One of four internal clocks (ϕ/8, ϕ/32, ϕ/128, ϕ/512) can be selected independently for each channel.
- Interrupt sources
  - A compare match interrupt can be requested independently for each channel.

17.1.2 Block Diagram

Figure 17.1 shows a block diagram of the CMT.

22.7 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes are made by setting the PSU1, ESU1, P1, E1, PV1, and EV1 bits in FLMCR1 for addresses H'00000–H'1FFFF, or the PSU2, ESU2, P2, E2, PV2, and EV2 bits in FLMCR2 for addresses H'20000–H'3FFFF.

The flash memory cannot be read while being programmed or erased. Therefore, the program (programming control program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU1, PSU1, EV1, PV1, E1, and P1 bits in FLMCR1, or the ESU2, PSU2, EV2, PV2, E2, and P2 bits in FLMCR2, is executed by a program in flash memory.
2. When programming or erasing, set FWP to low level (programming/erasing will not be executed if FWP is set to high level).
3. Programming should be performed in the erased state. Do not perform additional programming on previously programmed addresses.
4. Do not program addresses H'00000–H'1FFFF and H'20000–H'3FFFF simultaneously. Operation is not guaranteed if this is done.

22.7.1 Program Mode (n = 1 for Addresses H'0000–H'1FFFF, n = 2 for Addresses H'20000–H'3FFFF)

When writing data or programs to flash memory, the program/program-verify flowchart shown in figure 22.13 should be followed. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 32 bytes at a time.

Following the elapse of 10 µs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 32-byte program data is stored in the program data area and reprogram data area, and the 32-byte data in the program data area in RAM is written consecutively to the program address (the lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0). Thirty-two consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 32-byte data transfer must be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. Set a minimum value of 300 µs or more as the WDT overflow period. After this, preparation for program mode (program setup) is carried out by setting the PSUn bit in FLMCRn, and after the elapse of 50 µs or more, the operating mode is switched to program mode by setting the Pn bit in

22.8 Protection

There are two kinds of flash memory program/erase protection, hardware protection and software protection.

22.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2). The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained in the error-protected state. (See table 22.8.)

Table 22.8 Hardware Protection

		Function	
Item	**Description**	**Program**	**Erase**
FWP pin protection	• When a high level is input to the FWP pin, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a reset (including a WDT overflow reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

22.9 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses can be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 22.16 shows an example of emulation of real-time flash memory programming.

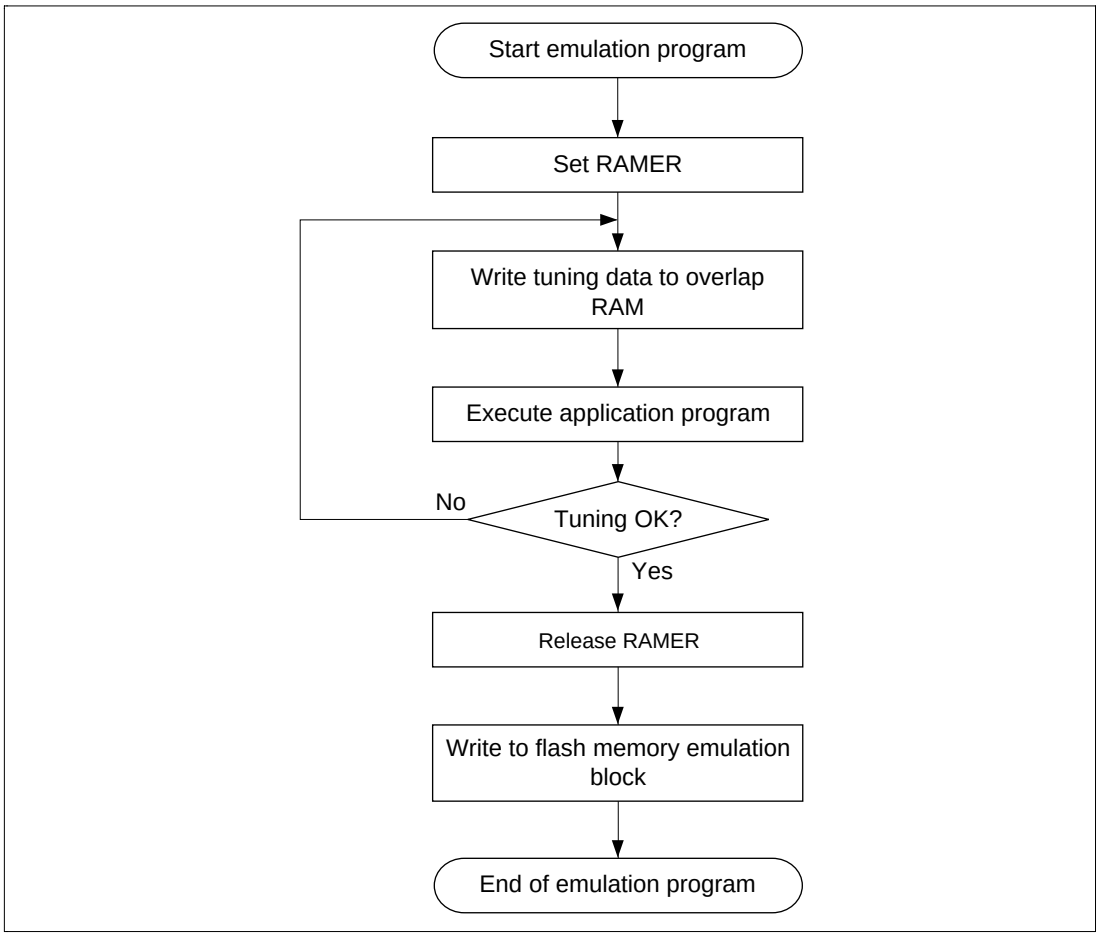

Figure 22.16 Flowchart for Flash Memory Emulation in RAM