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"

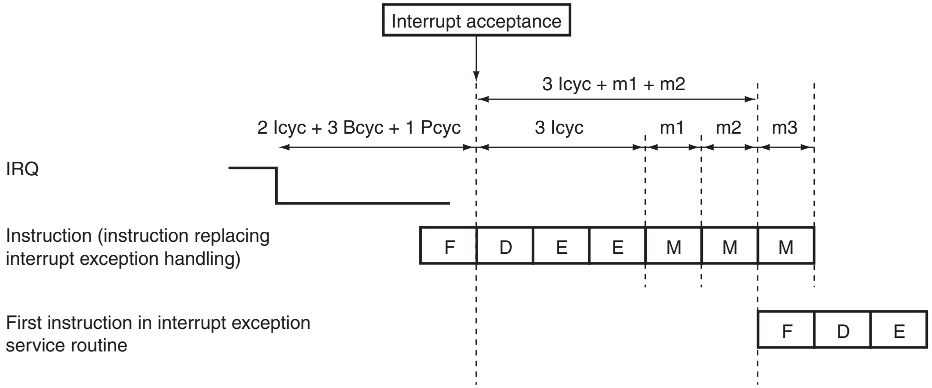
Details	
Product Status	Last Time Buy
Core Processor	SH2A-FPU
Core Size	32-Bit Single-Core
Speed	144MHz
Connectivity	I²C, SCI, SD, SIO, SPI, USB
Peripherals	DMA, POR, PWM, WDT
Number of I/O	115
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	640K x 8
Voltage - Supply (Vcc/Vdd)	1.1V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	208-LQFP
Supplier Device Package	208-LFQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5s72644w144fp-u0

Exception	Source	Timing of Source Detection and Start of Handling
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except immediately after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state).
	Slot illegal instructions	Starts from the decoding of undefined code placed directly after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state), of instructions that rewrite the PC, of 32-bit instructions, of the RESBANK instruction, of the DIVS instruction, or of the DIVU instruction.
	Integer division exceptions	Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by –1.
Instructions	FPU exceptions	Starts when detecting invalid floating point operation exception defined by IEEE standard 754, division-by-zero exception, overflow, underflow, or inexact exception. Also starts when qNaN or ±∞ is input to the source for a floating point operation instruction when the QIS bit in FPSCR is set.

When exception handling starts, the CPU operates as follows:

(1) Exception Handling Triggered by Reset

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 6.1.3, Exception Handling Vector Table, for more information. The vector base register (VBR) is then initialized to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized to 0. The BN bit in IBNR of the interrupt controller is also initialized to 0. The floating point status/control register (FPSCR) is initialized to H'00040001 by a power-on reset. The program begins running from the PC address fetched from the exception handling vector table.


Interrupt acceptance
3 Icyc + m1 + m2
2 Icyc + 3 Bcyc + 1 Pcyc
3 Icyc
m1
m2
m3
IRQ
Instruction (instruction replacing interrupt exception handling)
F
D
E
E
M
M
M
First instruction in interrupt exception service routine
F
D
E

[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)
F: Instruction fetch. Instruction is fetched from memory in which program is stored.
D: Instruction decoding. Fetched instruction is decoded.
E: Instruction execution. Data operation or address calculation is performed in accordance with the result of decoding.
M: Memory access. Memory data access is performed.

Figure 7.4 Example of Pipeline Operation when IRQ Interrupt is Accepted (No Register Banking)

8.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by means of MOV instructions. The instruction cache address array is mapped onto addresses H'F0000000 to H'F07FFFFF, and the data array onto addresses H'F1000000 to H'F17FFFFF. The operand cache address array is mapped onto addresses H'F0800000 to H'F0FFFFFF, and the data array onto addresses H'F1800000 to H'F1FFFFFF. Only longword can be used as the access size for the address array and data array, and instruction fetches cannot be performed.

8.4.1 Address Array

To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified.

In the address field, specify the entry address for selecting the entry, the W bit for selecting the way, and the A bit for specifying the existence of associative operation. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the address array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

The tag address, LRU bits, U bit (only for operand cache), and V bit are specified as data. Always specify 0 for the upper three bits (bits 31 to 29) of the tag address.

For the address and data formats, see figure 8.4.

The following three operations are possible for the address array.

(1) Address Array Read

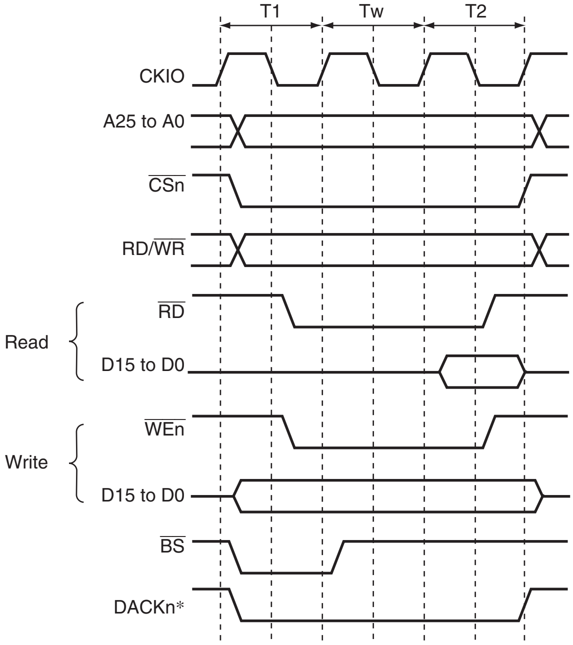
The tag address, LRU bits, U bit (only for operand cache), and V bit are read from the entry address specified by the address and the entry corresponding to the way. For the read operation, associative operation is not performed regardless of whether the associative bit (A bit) specified by the address is 1 or 0.

(2) Address-Array Write (Non-Associative Operation)

When the associative bit (A bit) in the address field is cleared to 0, write the tag address, LRU bits, U bit (only for operand cache), and V bit, specified by the data field, to the entry address specified by the address and the entry corresponding to the way. When writing to a cache line for which the U bit = 1 and the V bit =1 in the operand cache address array, write the contents of the cache line back to memory, then write the tag address, LRU bits, U bit, and V bit specified by the data field. When 0 is written to the V bit, 0 must also be written to the U bit of that entry. Write-

9.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for areas 1, 4, and 5 to insert wait cycles independently in read access and in write access. Areas 0, 2, 3, and 6 have common access wait for read cycle and write cycle. The specified number of Tw cycles are inserted as wait cycles in a normal space access shown in figure 9.8.



Note: * The waveform for DACKn is when active low is specified.

Figure 9.8 Wait Timing for Normal Space Access (Software Wait Only)

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 11.15.

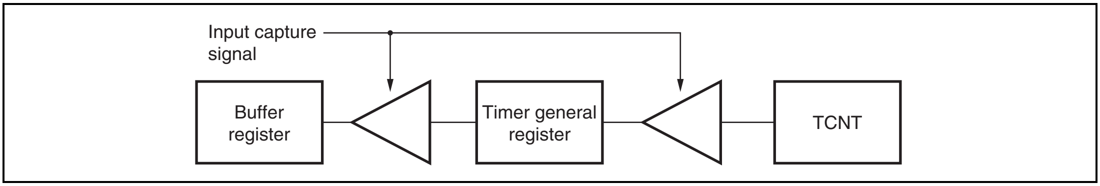
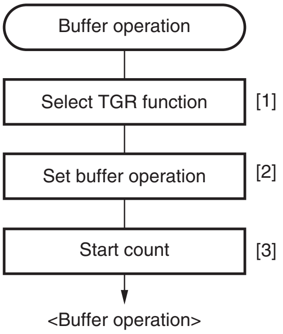


Figure 11.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 11.16 shows an example of the buffer operation setting procedure.



[1] Designate TGR as an input capture register or output compare register by means of TIOR.

[2] Designate TGR for buffer operation with bits BFA and BFB in TMDR.

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

Figure 11.16 Example of Buffer Operation Setting Procedure

(13) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode

Figure 11.127 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

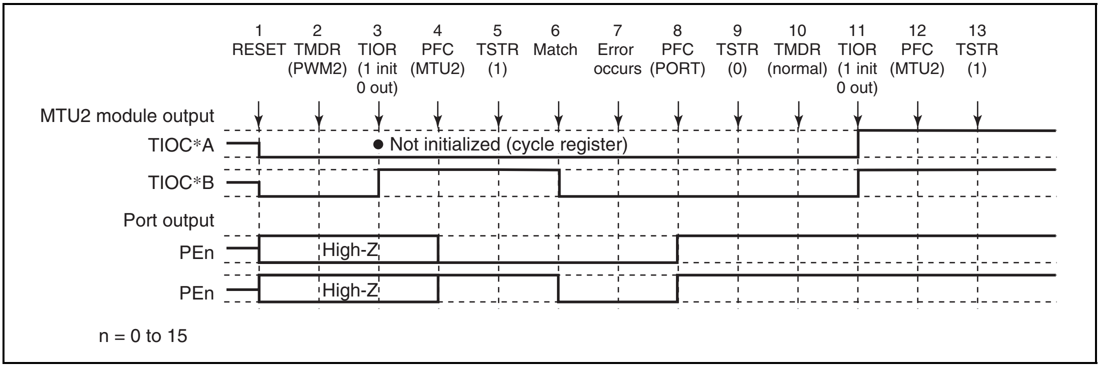
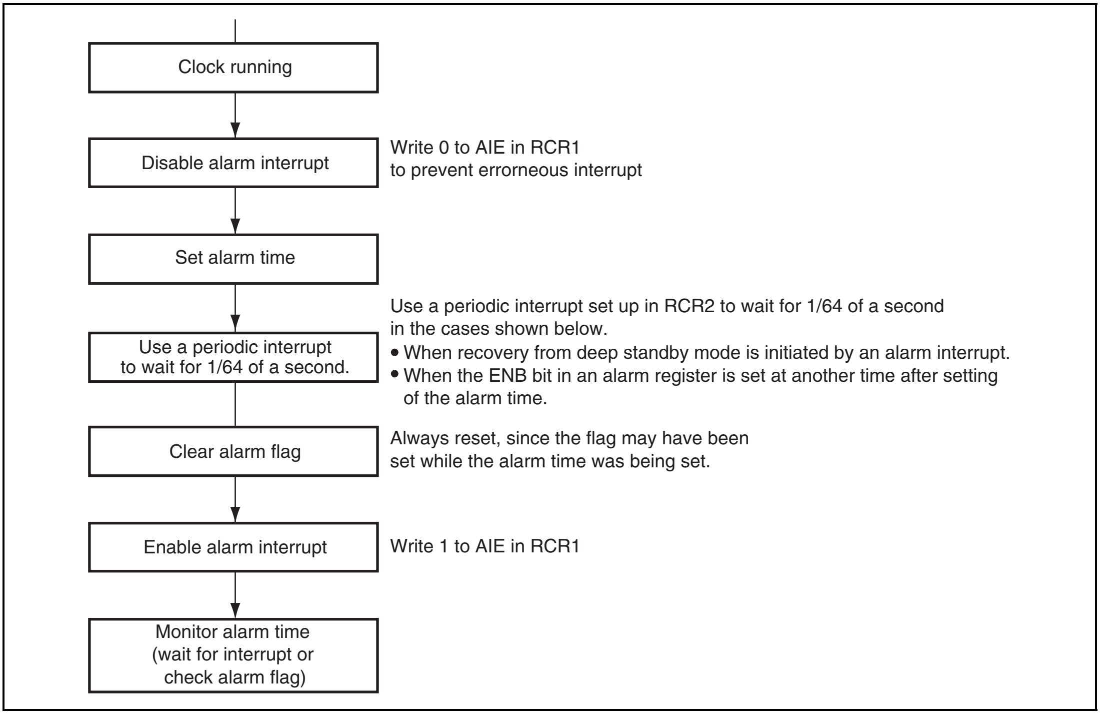


Figure 11.127 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, the module output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, TIOC *A is the cycle register.)
4. Set the multi-function timer pulse unit 2 output with the general I/O port.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the general I/O port and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set the multi-function timer pulse unit 2 output with the general I/O port.
13. Operation is restarted by TSTR.

14.4.4 Alarm Function

Figure 14.4 shows how to use the alarm function.


Clock running
Disable alarm interrupt
Set alarm time
Use a periodic interrupt to wait for 1/64 of a second.
Clear alarm flag
Enable alarm interrupt
Monitor alarm time (wait for interrupt or check alarm flag)

Write 0 to AIE in RCR1 to prevent errorneous interrupt

Use a periodic interrupt set up in RCR2 to wait for 1/64 of a second in the cases shown below.

- When recovery from deep standby mode is initiated by an alarm interrupt.
- When the ENB bit in an alarm register is set at another time after setting of the alarm time.

Always reset, since the flag may have been set while the alarm time was being set.

Write 1 to AIE in RCR1

Figure 14.4 Using Alarm Function

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, year, or any combination of these. Set the ENB bit in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is not placed to 0.

When the clock and alarm times match, 1 is set in the AF bit in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is set in the AIE bit in RCR1, an interrupt is generated when an alarm occurs.

The alarm flag is set when the clock and alarm times match. However, the alarm flag can be cleared by writing 0.

15.3.10 FIFO Data Count Set Register (SCFDR)

SCFDR is a 16-bit register which indicates the quantity of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR).

It indicates the quantity of transmit data in SCFTDR with the upper 8 bits, and the quantity of receive data in SCFRDR with the lower 8 bits. SCFDR can always be read by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	T[4:0]					-	-	-	R[4:0]				
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	T[4:0]	00000	R	T4 to T0 bits indicate the quantity of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and H'10 means that SCFTDR is full of transmit data.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	R[4:0]	00000	R	R4 to R0 bits indicate the quantity of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR full of receive data.

Figure 15.9 shows an example of the operation for reception.

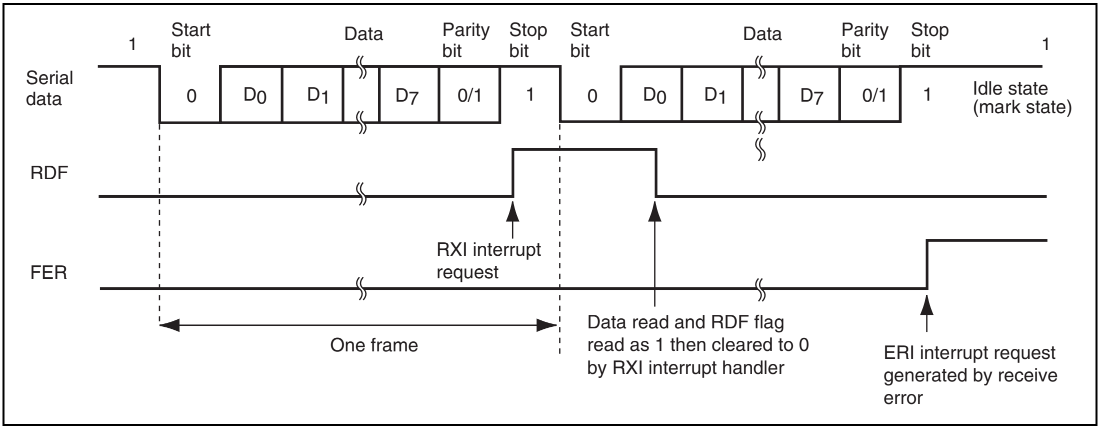


Figure 15.9 Example of Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

5. When modem control is enabled in channel 3, the $\overline{\text{RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that SCFRDR exceeds the number set for the RTS output active trigger.

Figure 15.10 shows an example of the operation when modem control is used.

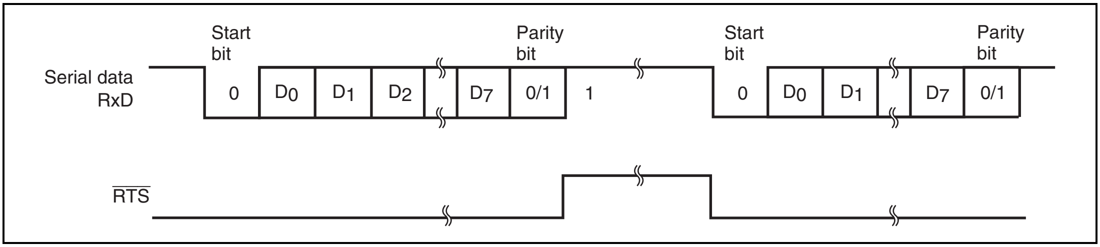


Figure 15.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

The above limitation is not applied to mailboxes which are not set as time-triggered transmission.

Important: Because of limitation on setting Tx-Trigger Time, only one Mailbox can be assigned to one time window.

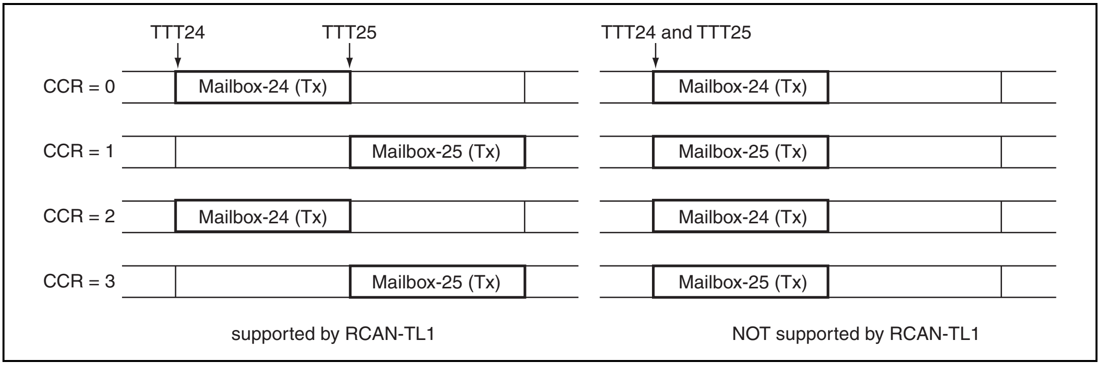


Figure 20.18 Limitation on Tx-Trigger Time

The value of TCMR2 as Watch_Trigger has to be larger than TTT(Mailbox-30), which shows the length of a basic cycle.

(3) Interpolated Sync Mode

In interpolated sync mode, synchronization is always driven by the internal counter after a sync code pattern has been detected at the start of decoding. Accordingly, this mode is effective when the sync patterns have been damaged.

However, decoding becomes incorrect after a change to the synchronization timing, since the change in timing is not followed.

Figure 23.12 shows the operation in interpolated sync mode.

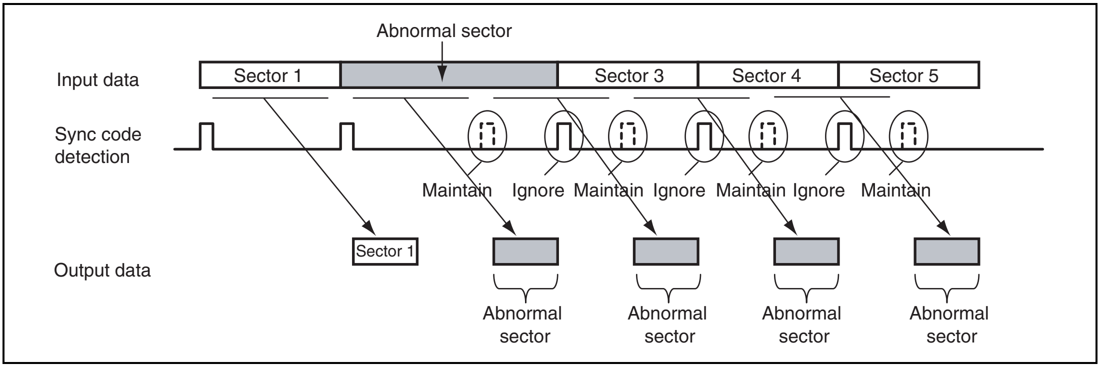


Figure 23.12 Operation in Interpolated Sync Mode

23.5.2 Timing of Status Registers Updates

The status information registers of the CD-ROM decoder are updated on each ISEC interrupt. The sector for which information is reflected in the status registers is selected by the ER0SEL bit of the CROMCTL4 register.

23.6 Usage Notes

23.6.1 Stopping and Resuming Buffering Alone during Decoding

When the data of the output stream are being not read out but operation of the CD-ROM decoder has continued until the buffers are full, the BUF_NG bit in the CBUFST2 register is set to 1; after that, the CD-ROM decoder becomes incapable of operation.

To stop buffering alone, clear the CBUF_EN bit in the CBUFCTL0 register to 0. If the automatic buffering function is in use, clear the CBUF_AUT in the CBUFCTL0 register to 0 at the same time. In this case, the sectors currently in the buffers must be read out.

To resume automatic buffering, set the CBUF_AUT and CBUF_EN bits in the CBUFCTL0 register at the same time.

23.6.2 When CROMST0 Status Register Bits are Set

1. When the ST_SECS bit in the CROMST0 register becomes set, stop decoding immediately and retry from one sector before the sector that was being decoded.
2. When the ST_SECL bit in the CROMST0 register becomes set, stop decoding immediately and retry from two sectors before the sector that was being decoded.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDYE	PIPE8 NRDYE	PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PIPE9NRDYE	0	R/W	NRDY Interrupt Enable for PIPE9 0: Interrupt output disabled 1: Interrupt output enabled
8	PIPE8NRDYE	0	R/W	NRDY Interrupt Enable for PIPE8 0: Interrupt output disabled 1: Interrupt output enabled
7	PIPE7NRDYE	0	R/W	NRDY Interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6NRDYE	0	R/W	NRDY Interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled
5	PIPE5NRDYE	0	R/W	NRDY Interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4NRDYE	0	R/W	NRDY Interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3NRDYE	0	R/W	NRDY Interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2NRDYE	0	R/W	NRDY Interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled

Register Name	Setting
VIDEO_VSTART[24:16]	For BT.656 NTSC: H'010 For BT.656 PAL: H'016 For BT.601: Specify the vertical start position of the valid video in the TOP field.
VIDEO_VSTART[8:0]	For BT.656 NTSC: H'117 For BT.656 PAL: H'14F For BT.601: Specify the vertical start position of the valid video in the BOTTOM field.
VIDEO_HSTART[8:0]	For BT.656 NTSC: H'114 For BT.656 PAL: H'120 For BT.601: Specify the horizontal start position of the valid video.
VIDEO_VSYNC_TIM1[25:16]	Refer to (1) Video Display Position and Register Settings described below.
VIDEO_VSYNC_TIM1[9:0]	Refer to (1) Video Display Position and Register Settings described below.
VIDEO_SAVE_NUM[9:0]	H'000
VIDEO_IMAGE_CNT[6:4]	Specify the vertical scaling ratio.
VIDEO_IMAGE_CNT[2:0]	Specify the horizontal scaling ratio.
VIDEO_BASEADR[31:0]	Specify the base address.
VIDEO_LINE_OFFSET[31:0]	Specify the line offset.
VIDEO_FIELD_OFFSET[31:0]	H'000
VIDEO_LINEBUFF_NUM[8:0]	Refer to (1) Video Display Position and Register Settings described below.
VIDEO_DISP_SIZE[24:16]	Specify the vertical size of the video to be displayed.
VIDEO_DISP_SIZE [9:0]	Specify the horizontal size of the video to be displayed.
VIDEO_DISP_HSTART	Specify the horizontal position of the video to be displayed.
SG_MODE [16]	H'1
VIDEO_VSYNC_TIM2	Refer to (1) Video Display Position and Register Settings described below.
VIDEO_MODE [1]	H'1
VIDEO_MODE [0]	H'1
GRCMEN2[0], [31]	H'1, H'1
GRCMEN1[0], [31]	H'1, H'1 (This setting starts display. Set this bit at the end of the procedure.)

32.2.8 Port C Control Registers 0 to 2 (PCCR0 to PCCR2)

PCCR0 to PCCR2 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port C.

(1) Port C Control Register 2 (PCCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PC10 MD0	-	-	-	PC9 MD0	-	-	PC8MD[1:0]	
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/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC10MD0	0	R/W	PC10 Mode Select the function of the PC10. 0: PC10 1: TIOC2B Note: This bit is reserved in the SH7262 Group. It is always read as 0. The write value should always be 0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC9MD0	0	R/W	PC9 Mode Select the function of the PC9. 0: PC9 1: TIOC2A Note: This bit is reserved in the SH7262 Group. It is always read as 0. The write value should always be 0.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(2) Port K Control Register 1 (PKCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PK7MD[1:0]		-	-	PK6MD[1:0]		-	-	PK5MD[1:0]		-	-	PK4MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PK7MD[1:0]	00	R/W	PK7 Mode Select the function of the PK7. 00: PK7 10: SD_CD 01: PWM1H 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PK6MD[1:0]	00	R/W	PK6 Mode Select the function of the PK6. 00: PK6 10: SD_WP 01: PWM1G 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PK5MD[1:0]	00	R/W	PK5 Mode Select the function of the PK5. 00: PK5 10: SD_D1 01: PWM1F 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PK4MD[1:0]	00	R/W	PK4 Mode Select the function of the PK4. 00: PK4 10: SD_D0 01: PWM1E 11: Setting prohibited

35.3.6 PWM Buffer Registers_nA, nC, nE, nG (PWBFR_nA, PWBFR_nC, PWBFR_nE, PWBFR_nG) (n = 1, 2)

There are four PWBFR_n registers (PWBFR_nA, PWBFR_nC, PWBFR_nE, and PWBFR_nG). When a PWCYR_n compare match occurs, data is transferred from the buffer register (PWBFR_n) to the duty register (PWDTR_n).

Bit:	15	14	13	12	11	10	9	8
	—	—	—	OTS	—	—	DT9	DT8
Initial Value:	1	1	1	0	1	1	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
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	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.
12	OTS	0	R/W	Output Terminal Select Holds the data to be sent to bit 12 in PWDTR_n.
11, 10	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.
9	DT9	0	R/W	Duty These bits hold the data to be sent to bits 9 to 0 in PWDTR_n.
8	DT8	0	R/W	
7	DT7	0	R/W	
6	DT6	0	R/W	
5	DT5	0	R/W	
4	DT4	0	R/W	
3	DT3	0	R/W	
2	DT2	0	R/W	
1	DT1	0	R/W	
0	DT0	0	R/W	

37.4.10 Serial Sound Interface Timing

Table 37.14 Serial Sound Interface Timing

Item	Symbol	Min.	Max.	Unit	Remarks	Figure
Output clock cycle	t_O	80	64000	ns	Output	Figure 37.53
Input clock cycle	t_I	80	64000	ns	Input	
Clock high	t_{HC}	32	—	ns	Bidirectional	
Clock low	t_{LC}	32	—	ns		
Clock rise time	t_{RC}	—	25	ns	Output	
Delay	t_{DTR}	–5	25	ns	Transmit	Figures 37.54, 37.55
Setup time	t_{SR}	25	—	ns	Receive	
Hold time	t_{HTR}	5	—	ns	Receive, transmit	

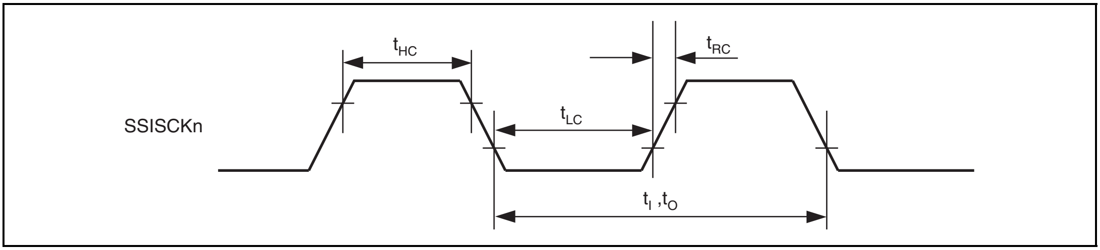


Figure 37.53 Clock Input/Output Timing

38.3 Handling of Pins in Deep Standby Mode

How pins are to be handled in deep standby mode is indicated below.

For the states of pins in deep standby mode, refer to the corresponding items under section, 38.1, Pin States. Handling of unused pins as described under section 38.2, Treatment of Unused Pins, also applies in deep standby mode.

Table 38.5 Handling of Pins in Deep Standby Mode

Pin	Handling
1.2-V power (Vcc, PLLVcc, USBDVcc, USBUVcc, USBAVcc)	Supply power at 1.2 V
3.3-V power (PVcc, AVcc, USBDPVcc, USBAPVcc)	Supply power at 3.3 V
Ground (Vss, PLLVss, USBDVss, USBUVss, USBAVss, AVss, USBDPVss, USBAPVss)	Connect to ground
VBUS	Fix the level on this pin (pull it up or down, or connect it to the power-supply or ground level) or open circuit. However, note that current as indicated in table 37.2, DC Characteristics (2) [Current Consumption] will be drawn by the pin fixed to the high level.
REFRIN	Connect this pin to USBAPVss via a 5.6 kΩ ± 1 % resistor
AVref	Fix the level on this pin (from 3.0 V to AVcc)
EXTAL, RTC_X1, USB_X1	Connect the pins to the crystal oscillator or the clock-input signal, or to a fixed level (pull them up or down, or connect them to the power-supply or ground level)
XTAL, RTC_X2, USB_X2	Connect the pins to the crystal oscillator or open circuit
Dedicated input pins other than those listed above	Fix the level on the pins (pull them up or down, or connect them to the power-supply or ground level).
Input/output pins (other than those listed above) in the input state	Fix the level on the pins (pull them up or down).
Input/output pins (other than those listed above) in the high-impedance state	Fix the level on the pins (pull them up or down) or open circuit.