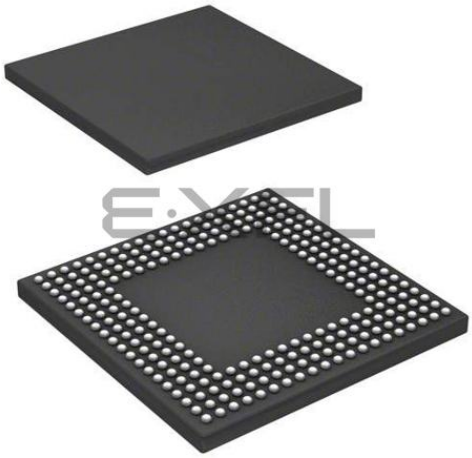
Renesas Electronics America Inc - D6417760BL200ADV Datasheet



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	Last Time Buy
Core Processor	SH-4
Core Size	32-Bit Single-Core
Speed	200MHz
Connectivity	Audio Codec, CANbus, EBI/EMI, FIFO, I²C, MFI, Memory Card, SCI, Serial Sound, SIM, SPI, USB
Peripherals	DMA, LCD, POR, WDT
Number of I/O	69
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	48K x 8
Voltage - Supply (Vcc/Vdd)	1.4V ~ 1.6V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	256-LFBGA
Supplier Device Package	256-LFBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/d6417760bl200adv

Item	Features
Serial communication interface (SCIF)	• Three full-duplex communications channels • On-chip 128-byte FIFOs for all channels • Choice of asynchronous mode or synchronous mode • Can select any bit rate generated by on-chip baud-rate generator • On-chip modem control function ($\overline{\text{SCIF\_RTS}}$ and $\overline{\text{SCIF\_CTS}}$) for channel 1 and 2
Audio codec interface (HAC)	• Digital interface for audio codec • Supports transfer for slot 1 to slot 4 • Choice of 16- or 20-bit DMA transfer • Supports various sampling rates by adjusting slot data • Generates interrupt: data ready, data request, overflow, and underrun
Serial sound interface (SSI)	• 2-channel bi-directional transfer (maximum) • Support multi-channel and compressed-data transfer • Selectable frame size
I^2C bus interface (I^2C)	• 2 channels (maximum) • Master/slave • 16-byte FIFO • Supports high-speed mode (400 kbits/sec) • Supports version 1.0
Multimedia card interface (MMCIF)	• Supports MMC mode • A maximum bit rate of 20 Mbps at 20 MHz of peripheral clock • Interface with MCCLK output for transfer clock output, MCCMD I/O for command output/response input, MCDAT I/O (data I/O) • Four interrupt sources
Smart card interface (SIM)	• Supports ISO/IEC7816-3 (Identification card) • Asynchronous half-duplex transfer (8 bits) • Can select any bit rate generated by on-chip baud-rate generator • Generates and checks parity bit • Four interrupt sources

- H'FC00 0000 to H'FFFF FFFF

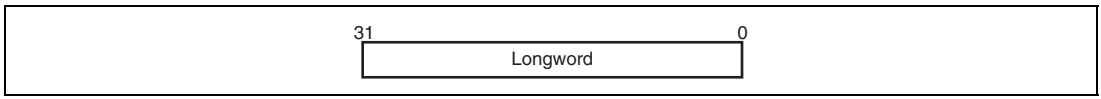
  Access to area H'FC00 0000 to H'FFFF FFFF in user mode will cause an address error. Memory-mapped registers can be referenced in user mode by means of access that involves address translation.

Note: Do not access addresses to which registers are not mapped in either area. The operation of an access to an address with no register mapped is undefined. Also, memory-mapped registers must be accessed using a fixed data size. The operation of an access using an invalid data size is undefined.

2.4 Data Formats

2.4.1 Data Format in Registers

Register operands are always longwords (32 bits). When a memory operand is only a byte (8 bits) or a word (16 bits), it is sign-extended into a longword when loaded into a register.



2.4.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in an 8-bit byte, 16-bit word, or 32-bit longword form. A memory operand less than 32 bits in length is sign-extended before being loaded into a register.

A word operand must be accessed starting from a word boundary (even address of a 2-byte unit: address 2n), and a longword operand starting from a longword boundary (even address of a 4-byte unit: address 4n). An address error will result if this rule is not observed. A byte operand can be accessed from any address.

Big endian or little endian byte order can be selected for the data format. The endian should be set with the MD5 external pin after a power-on reset. Big endian is selected when the MD5 pin is low, and little endian when high. The endian cannot be changed dynamically. Bit positions are numbered left to right from most-significant to least-significant. Thus, in a 32-bit longword, the leftmost bit, bit 31, is the most significant bit and the rightmost bit, bit 0, is the least significant bit.

The data format in memory is shown in figure 2.4.

Section 3 Floating-Point Unit (FPU)

3.1 Features

The FPU has the following features.

- Designed to meet IEEE754 standard
- 32 single-precision floating-point registers (can also be referenced as 16 double-precision registers)
- Two rounding modes: Round to Nearest and Round to Zero
- Two denormalization modes: Flush to Zero and Treat Denormalized Number
- Six exception sources: FPU Error, Invalid Operation, Divide By Zero, Overflow, Underflow, and Inexact
- Comprehensive instructions: Single-precision, double-precision, graphics support, and system control

When the FD bit in SR is set to 1, the FPU cannot be used, and an attempt to execute an FPU instruction will cause an FPU disable exception.

3.2 Data Formats

3.2.1 Floating-Point Format

A floating-point number consists of the following three fields:

- Sign (s)
- Exponent (e)
- Fraction (f)

The FPU can handle single-precision and double-precision floating-point numbers, using the formats shown in figures 3.1 and 3.2.

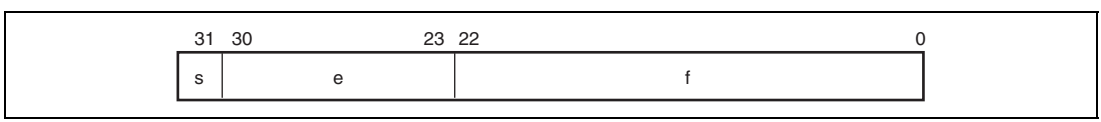


Figure 3.1 Format of Single-Precision Floating-Point Number

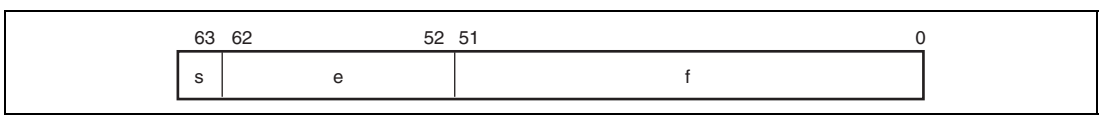


Figure 3.2 Format of Double-Precision Floating-Point Number

Table 3.2 Floating-Point Ranges

Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number	H'7F7F FFFF to H'0080 0000	H'7FEF FFFF FFFF FFFF to H'0010 0000 0000 0000
Positive denormalized number	H'007F FFFF to H'0000 0001	H'000F FFFF FFFF FFFF to H'0000 0000 0000 0001
Positive zero	H'0000 0000	H'0000 0000 0000 0000
Negative zero	H'8000 0000	H'8000 0000 0000 0000
Negative denormalized number	H'8000 0001 to H'807F FFFF	H'8000 0000 0000 0001 to H'800F FFFF FFFF FFFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 to H'FFEF FFFF FFFF FFFF
Negative infinity	H'FF80 0000	H'FFF0 0000 0000 0000
Quiet non-number	H'FF80 0001 to H'FFBF FFFF	H'FFF0 0000 0000 0001 to H'FFF7 FFFF FFFF FFFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 to H'FFFF FFFF FFFF FFFF

3.2.2 Non-Numbers (NaN)

Figure 3.3 shows the bit pattern of a non-number (NaN). A value is NaN in the following case:

- Sign bit: Don't care
- Exponent field: All bits are 1
- Fraction field: At least one bit is 1

The NaN is a signaling NaN (sNaN) if the MSB of the fraction field is 1, and a quiet NaN (qNaN) if the MSB is 0.

7.3.8 Prefetch Operation

This LSI supports a prefetch instruction to reduce the cache fill penalty incurred as the result of a cache miss. If it is known that a cache miss will result from a read or write operation, it is possible to fill the cache with data beforehand by means of the prefetch instruction to prevent a cache miss due to the read or write operation, and so improve software performance. If a prefetch instruction is executed for data already held in the cache, or if the prefetch address results in a UTLB miss or a protection violation, the result is no operation, and an exception is not generated. Details of the prefetch instruction are given in the Programming Manual.

- Prefetch instruction : PREF @Rn

7.3.9 Note on Using Cache Enhanced Mode

When cache enhanced mode (CCR.EMODE = 1) is specified and OC RAM mode (CCR.ORA = 1) is selected, in which half of the operand cache is used as internal RAM, internal RAM data may be updated incorrectly.

Conditions Under which Problem Occurs: Incorrect data may be written to RAM when the following four conditions are satisfied.

Condition 1: Cache enhanced mode (CCR.EMODE = 1) is specified.

Condition 2: The RAM mode (CCR.ORA = 1) in which half of the operand cache is used as internal RAM is specified.

Condition 3: An exception or an interrupt occurs.

Note: This includes a break triggered by a debugging tool swapping an instruction (a break occurring when a TRAPA instruction or undefined instruction code H'FFFD is swapped for an instruction).

Condition 4: A store instruction (MOV, FMOV, AND.B, OR, B, XOR.B, MOVCA.L, STC.L, or STS.L) that accesses internal RAM (H'7C000000 to H'7FFFFFFF) exists within four instructions after the instruction associated with the exception or interrupt described in condition 3. This includes cases where the store instruction that accesses internal RAM itself generates an exception.

Description: When the problem occurs, 8 bytes of incorrect data is written to the 8-byte boundary that includes an address that differs by H'2000 from the address accessed by the store instruction that accesses internal RAM mentioned in condition 4. For example, when a long word is stored at address H'7C000204, the 8 bytes of data in the internal RAM mapped to addresses H'7C002200 to H'7C002207 becomes corrupted.

9.5 Operation

9.5.1 Interrupt Operation Sequence

The sequence of operations when an interrupt is generated is described below. Figure 9.3 shows a flowchart of the interrupt operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the interrupt requests sent, according to the priority levels set in IPRA to IPRD and INTPRI00 to INTPRI0C. Lower-priority interrupts are held pending. If two of these interrupts have the same priority level, or if multiple interrupts occur within a single module, the interrupt with the highest priority according to table 9.4 is selected.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask level (IMASK3 to IMASK0) in SR of the CPU. If the request's priority level is higher than the level in bits IMASK3 to IMASK0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. The CPU accepts an interrupt between instructions.
5. The interrupt source code is set in the interrupt event register (INTEVT).
6. The contents of the status register (SR) and program counter (PC) are saved to SSR and SPC, respectively. The R15 contents at this time are saved in SGR.
7. The block bit (BL), mode bit (MD), and register bank bit (RB) in SR are set to 1.
8. The CPU jumps to the start address of the interrupt-related exception handling routine (the sum of the value set in the vector base register (VBR) and H'0000 0600).

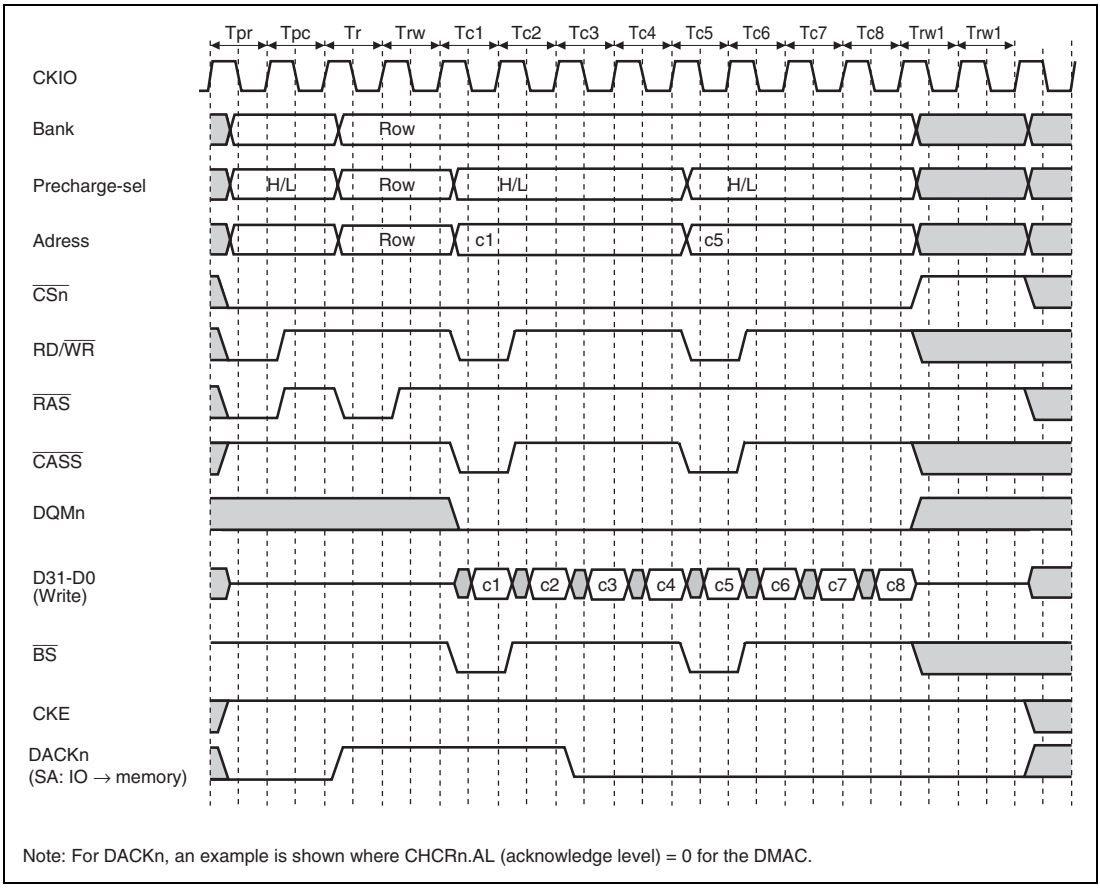
The exception handling routine may branch with the INTEVT value as its offset in order to identify the interrupt source. This enables it to easily branch to the handling routine for the particular interrupt source.

Notes: 1. In this LSI, the interrupt mask level bits (IMASK3 to IMASK0) in the status register (SR) of the CPU are not changed by acceptance of an interrupt.

2. Clear the interrupt source flag during the interrupt handling routine.

   To ensure that the cleared interrupt source is not inadvertently accepted again, read the interrupt source flag after it has been cleared, wait for the interval shown in table 9.8, and then clear the BL bit or execute an RTE instruction.

3. For some interrupt sources, the interrupt masks (INTMSK00 and INTMSK04) must be cleared using INTMSKCLR00 and INTMSKCLR04.



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.25 Burst Write Timing (Different Row Addresses)

(8) Pipelined Access

When the RASD bit in MCR is set to 1, pipelined access is performed for faster access to synchronous DRAM between an access by the CPU and an access by the DMAC or for consecutive accesses by the DMAC. Since synchronous DRAM is internally divided into two or four banks, after a READ or WRIT command is issued for one bank it is possible to issue a PRE, ACTV, or other command during the CAS latency cycle, data latch cycle, or data write cycle for shortening the access cycle.

When a read access is followed by another read access to the same row address, after a READ command has been issued, another READ command is issued before the end of the data latch cycle so that read data is on the data bus continuously. When an access is made to another row address and a different bank, the PRE command or ACTV command can be issued during the CAS latency cycle or data latch cycle. If there are consecutive access requests for different row

Bit	Bit Name	Initial Value	R/W	Description
0	DE	0	R/W	DMAC Enable Enables operation of the corresponding channel. When auto-request is specified by bits RS3 to RS0, setting this bit to 1 starts the transfer. When an external request or on-chip peripheral module request is generated, a transfer request after setting this bit to 1 starts the transfer. During a transfer, clearing this bit to 0 stops the transfer. The transfer enable state is not entered by setting this bit to 1 if the TE bit is set to 1, the DME bit in DMAOR is cleared to 0, or the NMIF or AE bit in DMAOR is set to 1. 0: Operation of corresponding channel is disabled 1: Operation of corresponding channel is enabled

16.3.5 Channels 0 to 3 Time Registers (CMTCH0T to CMTCH3T)

In output compare mode, these registers specify the value to be compared with the free-running timer. In input capture mode, this register stores the free-running timer values or the 16-bit timer values on the active edge of the input. Every time an edge is detected, these registers are updated and the new captured value will be saved.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel n time															
Initial value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel n time															
Initial value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.6 Channels 0 to 3 Stop Time Registers (CMTCH0ST to CMTCH3ST)

In output compare mode, these registers specify the value to be compared with the free-running timer. When this value is reached, the timer output is reset to the inactive state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel n stop time															
Initial value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel n stop time															
Initial value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.7 Channels 0 to 3 Counters (CMTCH0C to CMTCH3C)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel n counter															
Initial value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(1) Data Transfer Format

Table 17.6 shows the data transfer formats that can be used. Any of 8 transfer formats can be selected according to the SCSMR settings.

Table 17.6 Serial Transfer Formats (Asynchronous Mode)

SCSMR Settings			Serial Transfer Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	S	8-bit data								STOP		
0	0	1	S	8-bit data								STOP	STOP	
0	1	0	S	8-bit data								P	STOP	
0	1	1	S	8-bit data								P	STOP	STOP
1	0	0	S	7-bit data							STOP			
1	0	1	S	7-bit data							STOP	STOP		
1	1	0	S	7-bit data							P	STOP		
1	1	1	S	7-bit data							P	STOP	STOP	

Legend:
S : Start bit
STOP : Stop bit
P : Parity bit

Bit	Bit Name	Initial Value	R/W	Description
1	WAIT_ER	0	R/W	Wait Error Indicates the wait timer error status. 0: Indicates that the interval between the start of two successive characters has not exceeded the etu set by SIWAIT. [Clearing Conditions] • On reset • When 0 is written to WAIT_ER while its value is 1 1: Indicates that the interval between the start of two successive characters has exceeded the etu set by SIWAIT. [Setting Conditions] • In T = 0 mode, when the interval between the start of two successive characters exceeds the etu (value of 60 x SIWAIT: working wait time). • In T = 1 mode, when the interval between the start of two successive characters exceeds the etu (SIWAIT value: Guardtime). Notes: 1. Even if the RE bit in SISCR is cleared to 0, the WAIT_ER flag is unaffected, and the previous state is maintained. 2. In T = 0 mode, changing the RE bit from 0 to 1 may not set the WAIT_ER bit, even if the setting conditions for the WAIT_ER bit are satisfied. In this condition, the WAIT_ER bit is set at the timing of 60 × (SCWAIT + n) etu after the last transmission or reception. n is a whole number and it depends on the timing at which the RE bit is set. 3. In T = 0 mode, to avoid making the WAIT_ER bit set at the timing of 60 × (SCWAIT + n) etu after the last transmission or reception, the following procedure should be followed: Change the protocol bit (PB) in the smart card mode register (SISCMR) from 0 to 1 and again change the PB bit to 0. In T = 1 mode, to avoid making the WAIT_ER bit set at the timing of (SCWAIT) etu after the last reception, the following procedure should be followed: Change the PB bit in SISCMR from 1 to 0 and again change the PB bit to 1.

Bit	Bit Name	Initial Value	R/W	Description
9				• DWL = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 0 (left aligned) The data bits which are used in SSIRDR or SSITDR are the following: Bits 31 to (32 – number of bits having data word length specified by DWL). If DWL = 011 then data word length is 20 bits and bits 31 to 12 are used of either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 1 (right aligned) The data bits which are used in SSIRDR or SSITDR are the following: Bits (number of bits having data word length specified by DWL - 1) to 0. If DWL = 011 then data word length is 20 bits and bits 19 to 0 are used of either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 110 (data word length: 32 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus.
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSI_WS and SSI_SDATA 1: No delay between SSI_WS and SSI_SDATA This bit must be set to 1 when CPEN = 1. A one-clock cycle delay is not supported when the SSI module is configured to be a slave transmitter (SWSD = 0 and TRMD = 1). In this situation, this bit should be set to 0.

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Response register 8	RSPR8	R/W	H'FE50 0028	H'1E50 0028	8	Pck
Response register 9	RSPR9	R/W	H'FE50 0029	H'1E50 0029	8	Pck
Response register 10	RSPR10	R/W	H'FE50 002A	H'1E50 002A	8	Pck
Response register 11	RSPR11	R/W	H'FE50 002B	H'1E50 002B	8	Pck
Response register 12	RSPR12	R/W	H'FE50 002C	H'1E50 002C	8	Pck
Response register 13	RSPR13	R/W	H'FE50 002D	H'1E50 002D	8	Pck
Response register 14	RSPR14	R/W	H'FE50 002E	H'1E50 002E	8	Pck
Response register 15	RSPR15	R/W	H'FE50 002F	H'1E50 002F	8	Pck
Response register 16	RSPR16	R/W	H'FE50 0030	H'1E50 0030	8	Pck
Data timeout register	DTOUTR	R/W	H'FE50 0032	H'1E50 0032	16	Pck
Data register	DR	R/W	H'FE50 0040	H'1E50 0040	16	Pck
FIFO pointer clear register	FIFOCLR	W	H'FE50 0042	H'1E50 0042	8	Pck
DMA control register	DMACR	R/W	H'FE50 0044	H'1E50 0044	8	Pck
Interrupt control register 2	INTCR2	R/W	H'FE50 0046	H'1E50 0046	8	Pck
Interrupt status register 2	INTSTR2	R/W	H'FE50 0048	H'1E50 0048	8	Pck
Receive data timing select register	RDTIMSEL	R/W	H'FE50 004A	H'1E50 004A	8	Pck

Table 26.2 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Command register 0	CMDR0	H'00	H'00	Retained	*	Retained
Command register 1	CMDR1	H'00	H'00	Retained		Retained
Command register 2	CMDR2	H'00	H'00	Retained		Retained
Command register 3	CMDR3	H'00	H'00	Retained		Retained
Command register 4	CMDR4	H'00	H'00	Retained		Retained
Command register 5	CMDR5	H'00	H'00	Retained		Retained
Command start register	CMDSTRT	H'00	H'00	Retained		Retained
Operation control register	OPCR	H'00	H'00	Retained		Retained
Card status register	CSTR	H'0x	H'0x	Retained		Retained

Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Standby: Sleep by Sleep Instruction/ Deep Sleep	Standby: by Hardware	Standby: by Software/ Each Module
Interrupt control register 0	INTCR0	H'00	H'00	Retained	*	Retained
Interrupt control register 1	INTCR1	H'00	H'00	Retained		Retained
Interrupt status register 0	INTSTR0	H'00	H'00	Retained		Retained
Interrupt status register 1	INTSTR1	H'00	H'00	Retained		Retained
Transfer clock control register	CLKON	H'00	H'00	Retained		Retained
Command timeout control register	CTOCR	H'00	H'00	Retained		Retained
Transfer byte number count register	TBCR	H'00	H'00	Retained		Retained
Mode register	MODER	H'00	H'00	Retained		Retained
Command type register	CMDTYR	H'00	H'00	Retained		Retained
Response type register	RSPTYR	H'00	H'00	Retained		Retained
Response register 0	RSPR0	H'00	H'00	Retained		Retained
Response register 1	RSPR1	H'00	H'00	Retained		Retained
Response register 2	RSPR2	H'00	H'00	Retained		Retained
Response register 3	RSPR3	H'00	H'00	Retained		Retained
Response register 4	RSPR4	H'00	H'00	Retained		Retained
Response register 5	RSPR5	H'00	H'00	Retained		Retained
Response register 6	RSPR6	H'00	H'00	Retained		Retained
Response register 7	RSPR7	H'00	H'00	Retained		Retained
Response register 8	RSPR8	H'00	H'00	Retained		Retained
Response register 9	RSPR9	H'00	H'00	Retained		Retained
Response register 10	RSPR10	H'00	H'00	Retained		Retained
Response register 11	RSPR11	H'00	H'00	Retained		Retained
Response register 12	RSPR12	H'00	H'00	Retained		Retained
Response register 13	RSPR13	H'00	H'00	Retained		Retained
Response register 14	RSPR14	H'00	H'00	Retained		Retained
Response register 15	RSPR15	H'00	H'00	Retained		Retained
Response register 16	RSPR16	H'00	H'00	Retained		Retained
Data timeout register	DTOUTR	H'FFFF	H'FFFF	Retained		Retained
Data register	DR	H'xxxx	H'xxxx	Retained		Retained
FIFO pointer clear register	FIFOCLR	H'00	H'00	Retained		Retained

27.3.7 MFI Address Register (MFIADR)

The MFIADR is a 32-bit register which indicates the address in the MFRAM to be accessed by the external device via the MFI.

Specifying continuous access to the MFRAM in the LOCK bit in MFIMCR automatically performs auto-increment (+4) or auto-decrement (-4) of the address according to the AI/AD bit in MFIMCR, and updates MFIADR each time the external device accesses the MFRAM.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	A10	A9	A8	A7	A6	A5	A4	A3	A2	-	-
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/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 2	A10 to A2	All 0	R/W*	Address Specifies the memory space in the 2-Kbyte MFRAM to be accessed by the external device via the MFI, with 32-bit alignment.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * The external device can write to these bits via the MFI. The on-chip CPU cannot write to these bits.

30.3.8 LCDC Palette Control Register (LDPALCR)

LDPALCR selects whether the CPU or LCDC accesses the palette memory. When the palette memory is being used for display operation, display mode should be selected. When the palette memory is being written to, color-palette setting mode should be selected.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PALS	-	-	-	PALEN
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/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits always read as 0. The write value should always be 0.
4	PALS	0	R	Palette State Indicates the access right state of the palette. 0: Display mode: LCDC uses the palette 1: Color-palette setting mode: The host (CPU) uses the palette
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PALEN	0	R/W	Palette Read/Write Enable Requests the access right to the palette. 0: Request for transition to normal display mode 1: Request for transition to color palette setting mode

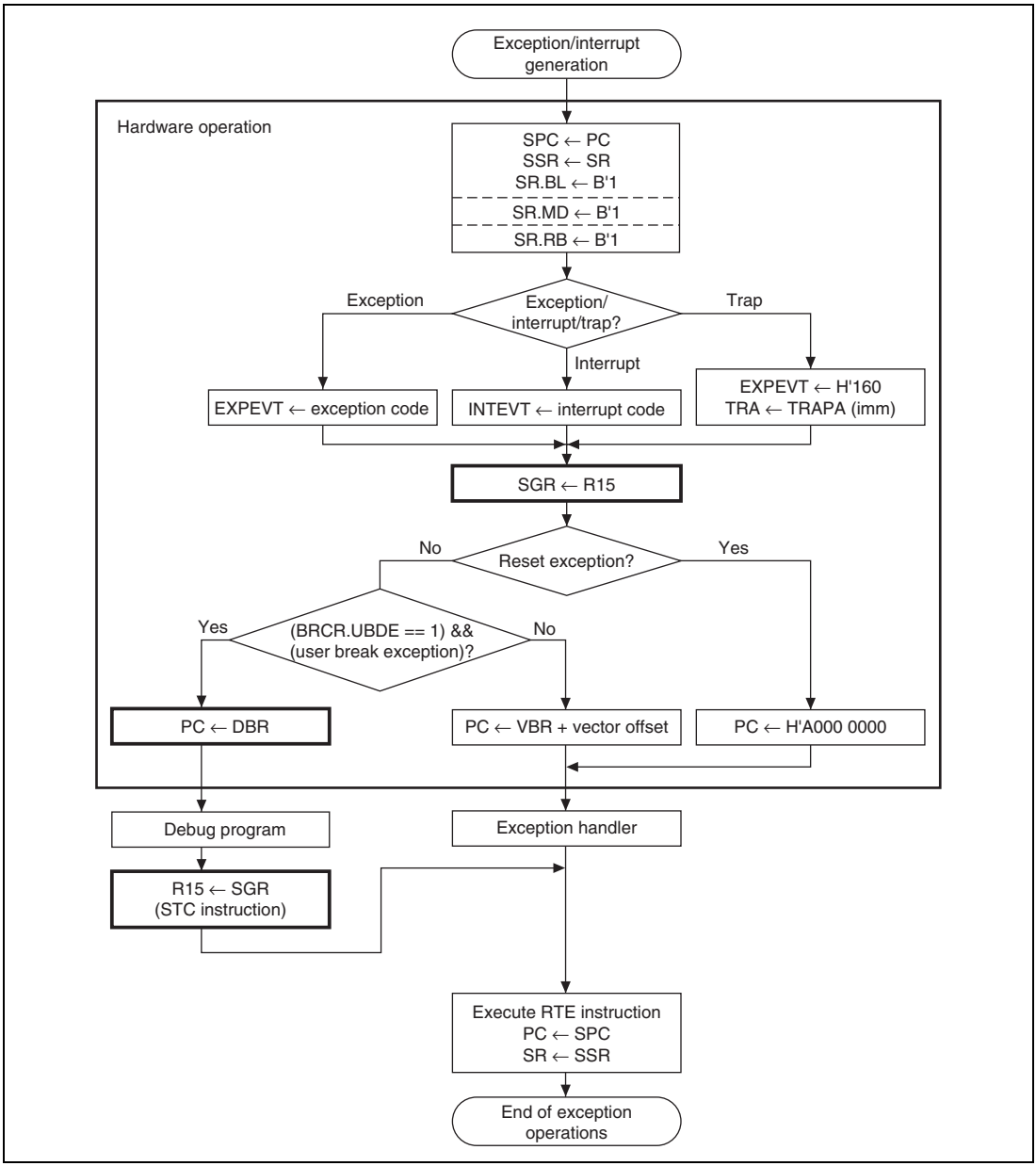


Figure 31.2 User Break Debug Support Function Flowchart

Module	Register Name	Abbrev.	P4 Address	Area 7 Address	Size	Sync Clock
SCIF Channel 1	Serial status register 1	SCFSR1	H'FE61 0010	H'1E61 0010	16	Pck
	Receive FIFO data register 1	SCFRDR1	H'FE61 0014	H'1E61 0014	8	Pck
	FIFO control register 1	SCFCR1	H'FE61 0018	H'1E61 0018	16	Pck
	Transmit FIFO data count register 1	SCTFDR1	H'FE61 001C	H'1E61 001C	16	Pck
	Receive FIFO data count register 1	SCRFDR1	H'FE61 0020	H'1E61 0020	16	Pck
	Serial port register 1	SCSPTR1	H'FE61 0024	H'1E61 0024	16	Pck
	Line status register 1	SCLSR1	H'FE61 0028	H'1E61 0028	16	Pck
	Serial error register 1	SCRER1	H'FE61 002C	H'1E61 002C	16	Pck
SCIF Channel.2	Serial mode register 2	SCSMR2	H'FE62 0000	H'1E62 0000	16	Pck
	Bit rate register 2	SCBRR2	H'FE62 0004	H'1E62 0004	8	Pck
	Serial control register 2	SCSCR2	H'FE62 0008	H'1E62 0008	16	Pck
	Transmit FIFO data register 2	SCFTDR2	H'FE62 000C	H'1E62 000C	8	Pck
	Serial status register 2	SCFSR2	H'FE62 0010	H'1E62 0010	16	Pck
	Receive FIFO data register 2	SCFRDR2	H'FE62 0014	H'1E62 0014	8	Pck
	FIFO control register 2	SCFCR2	H'FE62 0018	H'1E62 0018	16	Pck
	Transmit FIFO data count register 2	SCTFDR2	H'FE62 001C	H'1E62 001C	16	Pck
	Receive FIFO data count register 2	SCRFDR2	H'FE62 0020	H'1E62 0020	16	Pck
	Serial port register 2	SCSPTR2	H'FE62 0024	H'1E62 0024	16	Pck
	Line status register 2	SCLSR2	H'FE62 0028	H'1E62 0028	16	Pck
	Serial error register 2	SCRER2	H'FE62 002C	H'1E62 002C	16	Pck
SIM	Serial mode register	SISMR	H'FE48 0000	H'1E48 0000	8	Pck
	Bit rate register	SIBRR	H'FE48 0002	H'1E48 0002	8	Pck
	Serial control register	SISCR	H'FE48 0004	H'1E48 0004	8	Pck
	Transmit data register	SITDR	H'FE48 0006	H'1E48 0006	8	Pck
	Serial status register	SISSR	H'FE48 0008	H'1E48 0008	8	Pck
	Receive data register	SIRDR	H'FE48 000A	H'1E48 000A	8	Pck
	Smart card mode register	SISCMR	H'FE48 000C	H'1E48 000C	8	Pck
	Serial control 2 register	SISC2R	H'FE48 000E	H'1E48 000E	8	Pck
	Wait time register	SIWAIT	H'FE48 0010	H'1E48 0010	16	Pck
	Guard extension register	SIGRD	H'FE48 0012	H'1E48 0012	8	Pck
	Sampling register	SISMPL	H'FE48 0014	H'1E48 0014	16	Pck

Abbrev.	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/ 1	Bit 24/ 16/8/0	Module
CCR	EMODE	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	—	
	IIX	—	—	—	ICI	—	—	ICE	
	OIX	—	ORA	—	OCI	CB	WT	OCE	
QACR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	AREA0	AREA0	AREA0	—	—	
QACR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	AREA1	AREA1	AREA1	—	—	
TRA	—	—	—	—	—	—	—	—	Exception handling
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	imm	imm	
	imm	imm	imm	imm	imm	imm	—	—	
EXPEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
ICR	NMIL	MAI	—	—	—	—	NMIB	NMIE	INTC
	IRLM	—	—	—	—	—	—	—	
IPRA	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRB	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRC	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRD	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510