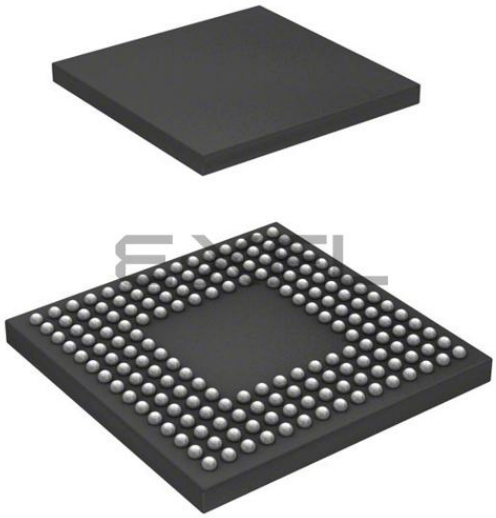

Welcome to E-XFL.COM

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	Active
Core Processor	H8S/2000
Core Size	16-Bit
Speed	26MHz
Connectivity	I²C, SCI
Peripherals	POR, PWM, WDT
Number of I/O	104
Program Memory Size	384KB (384K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	24K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	176-LFBGA
Supplier Device Package	176-LFBGA (13x13)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2551br26dv

Section 17 IEBus™ Controller (IEB) [H8S/2552 Group]

Section 18 Controller Area Network (HCAN) [H8S/2556 Group]

Section 20 Flash Memory

Section 18 Controller Area Network (HCAN) [H8S/2556 Group]

Section 20 Flash Memory

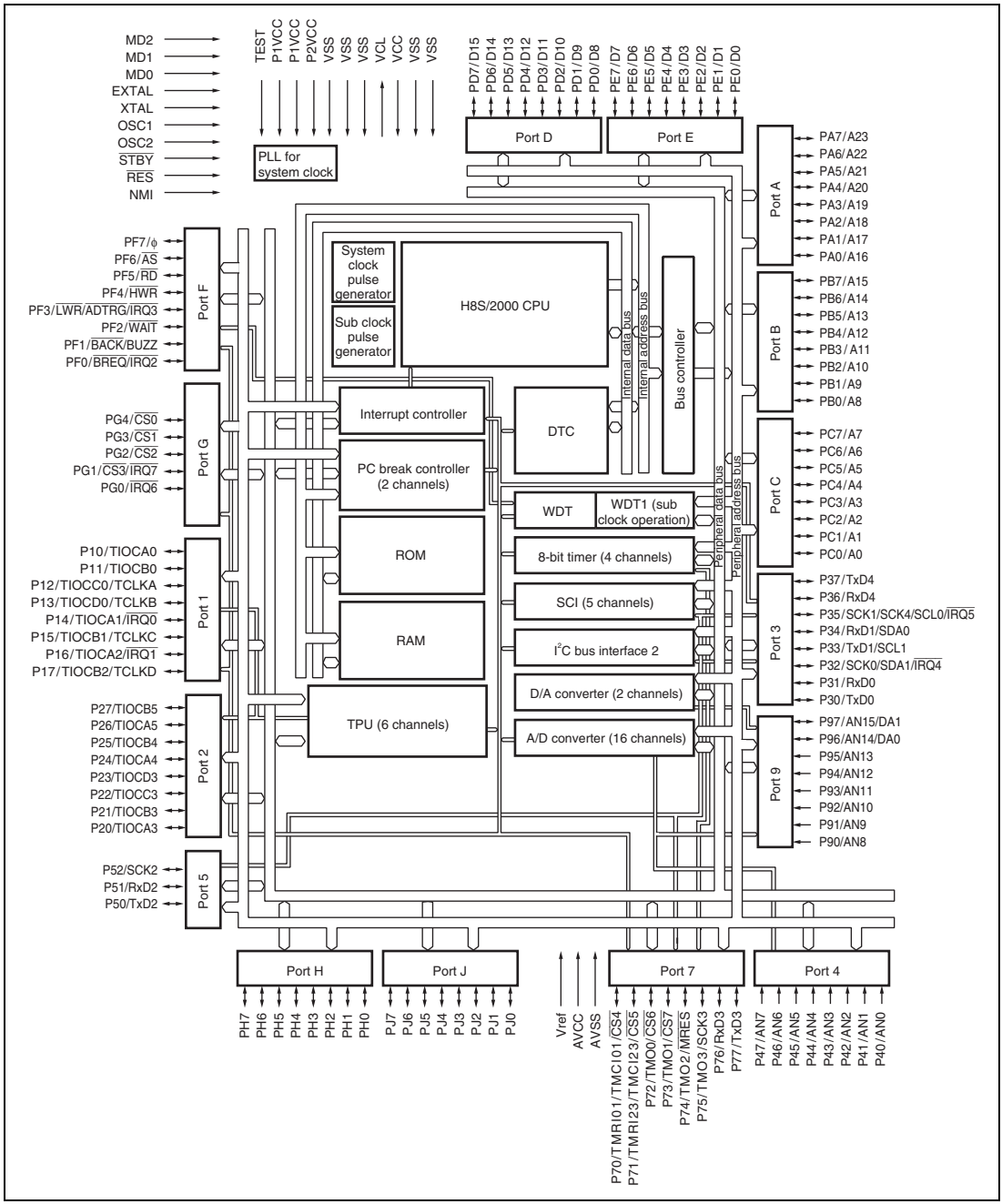

Figure 1.3 Internal Block Diagram of H8S/2506 Group

3.3 Operating Mode

3.3.1 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is valid. Immediately after a reset, ports A, B, and C become input ports. The AE3 to AEO bits in PFCR allow enable/disable setting of the address (A23 to A8) output, regardless of the corresponding DDR value. The pin which is disabled of the address output at ports A and B becomes an output port when the corresponding DDR is set to 1.

The address (A7 to A0) is output when the corresponding DDR is set to 1 at port C.

Ports D and E are data buses, and a part of the port F is the bus control signal.

Immediately after a reset, 8-bits bus mode is set and all the areas become 8-bit access space. However, when any of the areas is set to 16-bit access space by the bus controller, 16-bit bus mode is set and port E becomes the data bus.

3.3.2 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is valid and the external address space cannot be accessed.

All the I/O port can be used as an input/output port.

3.3.3 Pin Functions

Table 3.2 shows the pin functions in modes 6 and 7.

7.6 Basic Bus Interface

The basic bus interface enables direct connection of ROM, SRAM, and so on.

7.6.1 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function. When accessing external address space, it controls whether the upper data bus (D15 to D8) or lower data bus (D7 to D0) is used, according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-Bit Access Space: Figure 7.14 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word transfer instruction is performed as two-byte accesses, and a longword transfer instruction, as four-byte accesses.

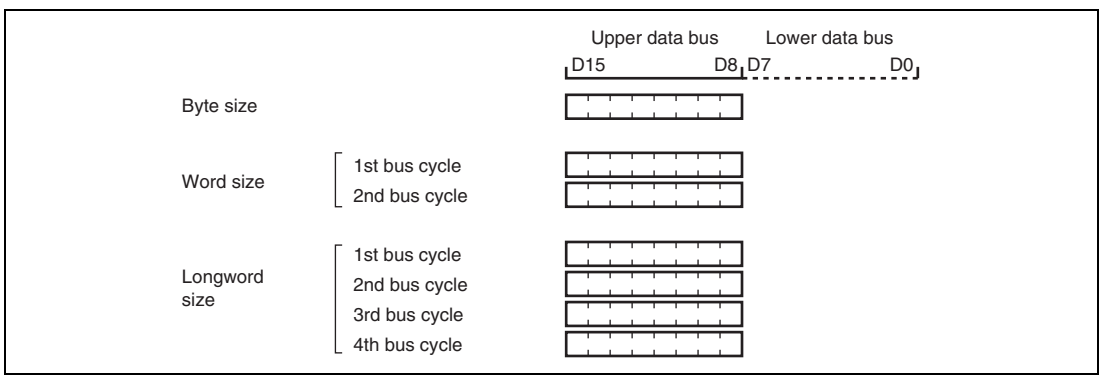

Figure 7.14 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space: Figure 7.15 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword transfer instruction is performed as two-word accesses.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.

8.2.1 DTC Mode Register A (MRA)

MRA selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SM1	Undefined	—	Source Address Mode 1 and 0
6	SM0	Undefined	—	These bits specify an SAR operation after a data transfer. 0X: SAR is fixed 10: SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1 and 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data transfer. 0X: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode 1 and 0
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: —
1	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode. 0: Destination side is repeat area or block area 1: Source side is repeat area or block area

9.13.2 Port F Data Register (PFDR)

PFDR stores output data for port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DR*	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose output port.
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

Note: * The value of PF7DR is not output on pin PF7 when the PF7DDR bit is set to 1 since the ϕ signal is output.

9.13.3 Port F Register (PORTF)

PORTF shows port F pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	—*	R	If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read. If a port F read is performed while PFDDR bits are cleared to 0, the pin states are read.
6	PF6	—*	R	
5	PF5	—*	R	
4	PF4	—*	R	
3	PF3	—*	R	
2	PF2	—*	R	
1	PF1	—*	R	
0	PF0	—*	R	

Note: * Determined by the states of pins PF7 to PF0.

Table 10.14 TIOR_1

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	X		Capture input source is TIOCB1 pin Input capture at both edges
	1	X	X		TGRC_0 compare match/ input capture Input capture at generation of TGRC_0 compare match/input capture

Legend:

X: Don't care

13.4 Operation in Asynchronous Mode

Figure 13.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer. In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfer data is latched at the center of each bit.

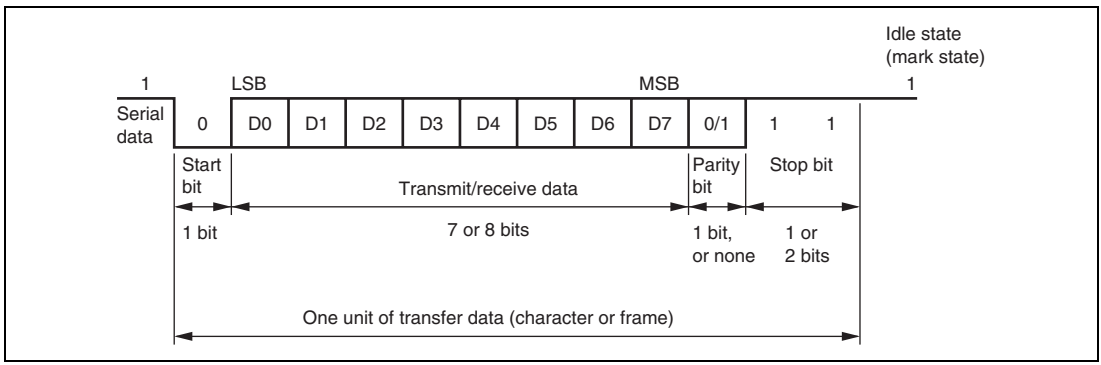

Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

13.4.1 Data Transfer Format

Table 13.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, see section 13.5, Multiprocessor Communication Function.

13.7.6 Serial Data Transmission (Except for Block Transfer Mode)

As data transmission in Smart Card interface mode involves error signal sampling and retransmission processing, the operations are different from those in normal serial communication interface mode (except for block transfer mode). Figure 13.26 illustrates the retransfer operation when the SCI is in transmit mode.

1. If an error signal is sent back from the receiving end after transmission of one frame is complete, the ERS bit in SSR is set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The ERS bit in SSR should be cleared to 0 by the time the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame in which an error signal indicating an abnormality is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set. Transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interrupt request is generated. Writing transmit data to TDR transfers the next transmit data.

Figure 13.28 shows a flowchart for transmission. A sequence of transmit operations can be performed automatically by specifying the DTC to be activated with a TXI interrupt source. In a transmit operation, the TDRE flag is set to 1 at the same time as the TEND flag in SSR is set, and a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI request is designated beforehand as a DTC activation source, the DTC will be activated by the TXI request, and transfer of the transmit data will be carried out. At this moment, when the DISEL bit in DTC is 0 and the transfer counter is other than 0, the TDRE and TEND flags are automatically cleared to 0 when data is transferred by the DTC. When the DISEL bit in the corresponding DTC is 1, or both the DISEL bit and the transfer counter are 0, flags are not cleared although the transfer data is written to TDR by DTC. Consequently, give the CPU an instruction of flag clear processing. In addition, in the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC is not activated. Therefore, the SCI and DTC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC, it is essential to set and enable the DTC before carrying out SCI setting. For details on the DTC setting procedures, see section 8, Data Transfer Controller (DTC).

15.3 Register Descriptions

The A/D converter has the following registers. For details on the module stop control register, see section 22.1.2, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

15.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

There are four 16-bit read-only ADDR registers; ADDRA to ADDRD, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each analog input channel, are shown in table 15.2.

The converted 10-bit data is stored in bits 6 to 15. The lower 6 bits are always read as 0.

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. Therefore, when reading ADDR, read only the upper byte, or read in word unit.

Table 15.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				
Channel Set 0 (CH3 = 0)		Channel Set 1 (CH3 = 1)		A/D Data Register to be Stored Results of A/D Conversion
Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	
AN0	AN4	AN8	AN12	ADDRA
AN1	AN5	AN9	AN13	ADDRB
AN2	AN6	AN10	AN14	ADDRC
AN3	AN7	AN11	AN15	ADDRD

15.3.2 A/D Control/Status Register (ADCSR)

ADCSR controls A/D conversion operations.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all specified channels in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DTC is activated by an ADI interrupt, and the DISEL bit in DTC is 0 with the transfer counter other than 0
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) request enabled when 1 is set
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters the wait state. Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when conversion on the specified channel is complete. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, or a transition to software standby mode, hardware standby mode, or module stop mode. The ADST bit can be set to 1 by software, a timer conversion start trigger, or the A/D external trigger input pin ($\overline{\text{ADTRG}}$).

Section 17 IEBus™ Controller (IEB) [H8S/2552 Group]

This LSI has an on-chip one-channel IEBus controller (IEB). The Inter Equipment Bus™ (IEBus™)*[1] is a small-scaled digital data transfer system for inter equipment data transfer.

This LSI does not have an on-chip IEBus driver/receiver, so it is necessary to mount a dedicated driver/receiver*[2] externally.

Notes: 1. IEBus is a trademark of NEC Electronics Corporation.
2. Bus interface driver/receiver IC: HA12187FP is recommended.

17.1 Features

- IEBus protocol control (layer 2) supported
  - Half duplex asynchronous communications
  - Multi-master system
  - Broadcast communications function
  - Selectable mode (three types) with different transfer speeds
- Data transfer by the data transfer controller (DTC)
  - Transfer buffer: 1 byte
  - Reception buffer: 1 byte
  - Up to 128 bytes of consecutive transfer/reception (maximum number of transfer bytes in mode 2)
- Operating frequency
  - 12 MHz, 12.58 MHz (IEB uses 1/2 divided external clock.)
  - 18 MHz, 18.87 MHz (IEB uses 1/3 divided external clock.)
  - 24 MHz, 25.16 MHz (IEB uses 1/4 divided external clock.)

Note: When selected communications mode 0 or mode 1 (± 1.5 %)
When selected communications mode 2 (± 0.5 %)

- Noise resistance is improved by mounting the IEBus driver/receiver (layer 1) externally.
- Module stop mode can be set.

Bit	Bit Name	Initial Value	R/W	Description
3	LCK	0	R	Lock Status Indication Set to 1 when a unit is locked by a lock request from the master unit. IELA1 and IELA2 values are valid only when this flag is set to 1. 1: A unit is locked [Setting condition] • When data for the number of bytes specified by the message length is not received after the control bits that make the unit locked are received from the master unit. (The LCK flag is set to 1 only when the message length exceeds the maximum number of transfer bytes in one frame. This flag is not set by completion of other errors.) 0: A unit is unlocked [Clearing condition] • When an unlock condition is satisfied or when an unlock command is issued.
2	—	0	R	Reserved This bit is always read as 0.
1	RSS	0	R	Receive Broadcast Bit Status Indicates the received broadcast bit value. This flag is valid when the slave/broadcast reception is started. (This flag is changed at the timing of setting the RxS flag in IERSR.) The previous value remains unchanged until the next slave/broadcast reception is started.

- Clearing by software
- Clearing by CAN bus operation

Eleven recessive bits must be received after HCAN sleep mode is cleared before CAN bus communication is re-enabled.

Clearing by software: HCAN sleep mode is cleared by writing a 0 to MCR5 from the CPU.

Clearing by CAN bus operation: The cancellation method is selected by the MCR7 bit setting in MCR. Clearing by CAN bus operation occurs automatically when the CAN bus performs an operation and this change is detected. In this case, the first message is not stored in a mailbox; messages will be received normally from the second message onward. When a change is detected on the CAN bus in HCAN sleep mode, the bus operation interrupt flag (IRR12) is set in the interrupt register (IRR). If the bus interrupt mask (IMR12) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

18.4.6 HCAN Halt Mode

The HCAN halt mode is provided to enable mailbox settings to be changed without performing an HCAN hardware or software reset. Figure 18.14 shows a flowchart of the HCAN halt mode.

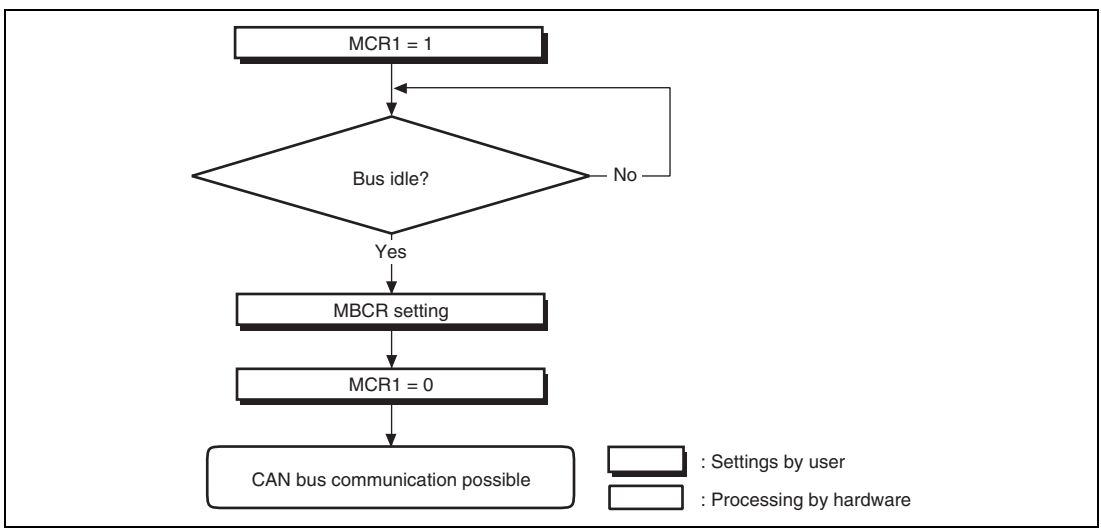

Figure 18.14 HCAN Halt Mode Flowchart

18.8.9 Usage of Bit Change Instructions

Do not use bit change instructions to clear flags, because the status flags of HCAN is cleared by writing 1. To clear a flag, use MOV instruction to write 1 to only the bits to be cleared.

18.8.10 HCAN TXCR Operation

1. When the transmit wait cancel register (TXCR) is used to cancel a transmit wait message in a transmit wait mailbox, the corresponding bit to TXCR and the transmit wait register (TXPR) may not be cleared even if transmission is canceled. This occurs when the following conditions are all satisfied.

- The HRxD pin is stacked to 1 because of a CAN bus error, etc.
- There is at least one mailbox waiting for transmission or being transmitted.
- The message transmission in a mailbox being transmitted is canceled by TXCR.
  If this occurs, transmission is canceled. However, since TXPR and TXCR states are indicated wrongly that a message is being cancelled, transmission cannot be restarted even if the stack state of the HRxD pin is canceled and the CAN bus recovers the normal state. If there are at least two transmission messages, a message which is not being transmitted is canceled and a message being transmitted retains its state.

To avoid this, one of the following countermeasures must be executed.

- Transmission must not be canceled by TXCR. When transmission is normally completed after the CAN bus has recovered, TXPR is cleared and the HCAN recovers the normal state.
- To cancel transmission, the corresponding bit to TXCR must be written to 1 continuously until the bit becomes 0. TXPR and TXCR are cleared and the HCAN recovers the normal state.

2. When the bus-off state is entered while TXPR is set and the transmit wait state is entered, the internal state machine does not operate even if TXCR is set during the bus-off state. Therefore transmission cannot be canceled. The message can be canceled when one message is transmitted or a transmission error occurs after the bus-off state is recovered. To clear a message after the bus-off state is recovered, the following countermeasure must be executed.

- A transmit wait message must be cleared by resetting the HCAN during the bus-off period.
  To reset the HCAN, the module stop bit (MSTPC2 in MSTPCRC) must be set or cleared. In this case, the HCAN is entirely reset. Therefore the initial settings must be made again.

Bit	Bit Name	Initial Value	R/W	Description
5	EE	—	R/W	Erasure Execution Error Detect 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed at the return from the user branch processing. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased. Erasing of the user boot MAT should be performed in boot mode or programming mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of FKEY value before start of the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other than H'5A)
3	EB	—	R/W	Erase Block Select Error Detect Returns the check result whether the specified erase-block number is in the block range of the user MAT. 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal
2, 1	—	—	—	Reserved Return 0.
0	SF	—	R/W	Success/Fail Indicates whether the erasing processing is ended normally or not. 0: Erasure is ended normally (no error) 1: Erasure is ended abnormally (error occurs)

Bit	Bit Name	Initial Value	R/W	Description
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

20.3.5 Flash Vector Address Data Register (FVADR)

This is a register to store the vector data when the flash vector address control register (FVACR) is used to enable the function to select the space where the vector table data is read. This register consists of four 8-bit registers: FVADRR, FVADRE, FVADRH, and FVADRL. This register is initialized to H'00000000 at a power-on reset or in hardware standby mode.

- FVADRR

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R/W	Set the vector address.

- FVADRE

Bit	Bit Name	Initial Value	R/W	Description
23 to 16	—	All 0	R/W	Set the vector address.

- FVADRH

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R/W	Set the vector address.

- FVADRL

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R/W	Set the vector address.

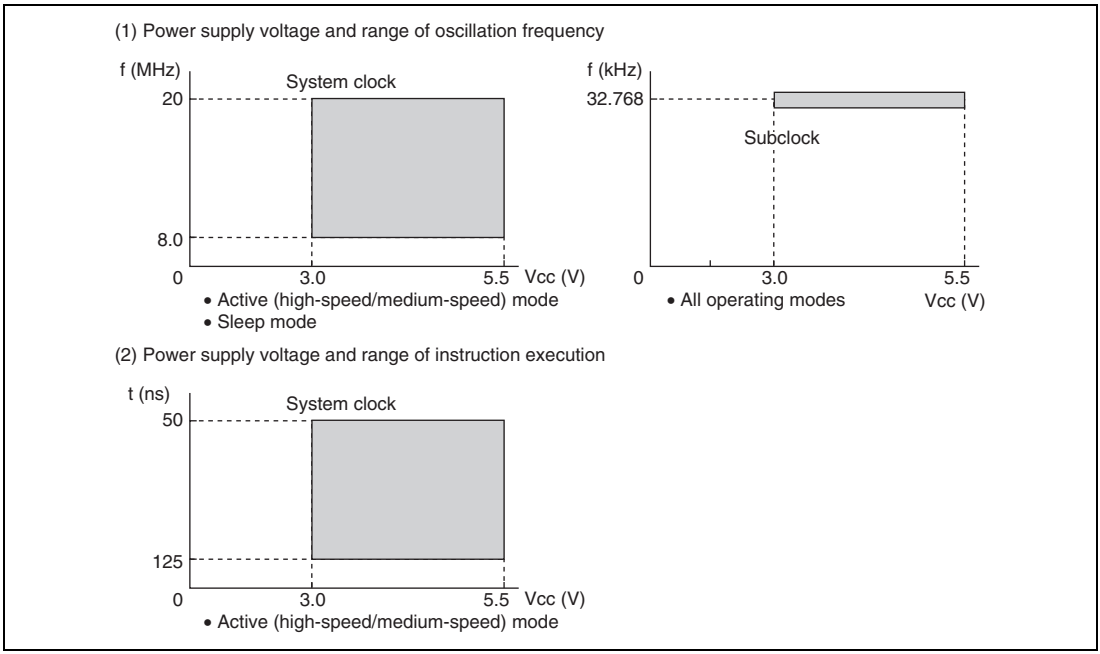

Figure 24.1 (2) Power Supply Voltage and Operating Ranges (H8S/2556 Group)

Item	Page	Revision (See Manual for Details)
20.4.4 Procedure Program and Storable Area for Programming Data Table 20.9 (3) Useable Area for Programming in User Boot Mode	723	Table amended

	Storable/Executable Area			Selected MAT	
Item	On-Chip RAM	User Boot MAT	User MAT	User Boot MAT	Embedded Program Storage Area
Execution of Initialization	○	×		○	
Determination of Initialization Result	○	○		○	
Operation for Initialization Error	○	○		○	
NMI Handling Routine	○	×		○	
Operation for Interrupt Inhibit	○	○		○	
Switching MATs by FMATS	○	×	○		
Operation for Writing H'5A to FKEY	○	×	○		
Operation for Settings of Program Parameter	○	×	○		
Execution of Programming	○	×	○		
Determination of Program Result	○	×	○		
Operation for Program Error	○	×※2	○		
Operation for FKEY Clear	○	×	○		
Switching MATs by FMATS	○	×		○	