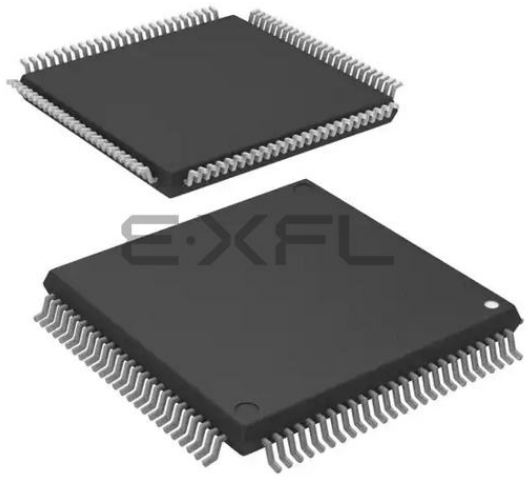


Welcome to E-XFL.COM

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	SH-2
Core Size	32-Bit Single-Core
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, SCI, SSU
Peripherals	POR, PWM, WDT
Number of I/O	57
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LFQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df71324ad80fpv

2.5.3 Arithmetic Operation Instructions

Table 2.12 Arithmetic Operation Instructions

Instruction		Operation	Code	Execution Cycles	T Bit
ADD	Rm,Rn	Rn + Rm → Rn	0011nnnnmmmm1100	1	—
ADD	#imm,Rn	Rn + imm → Rn	0111nnnniiiiiiii	1	—
ADDC	Rm,Rn	Rn + Rm + T → Rn, Carry → T	0011nnnnmmmm1110	1	Carry
ADDV	Rm,Rn	Rn + Rm → Rn, Overflow → T	0011nnnnmmmm1111	1	Overflow
CMP/EQ	#imm,R0	If R0 = imm, 1 → T	10001000iiiiiiii	1	Comparison result
CMP/EQ	Rm,Rn	If Rn = Rm, 1 → T	0011nnnnmmmm0000	1	Comparison result
CMP/HS	Rm,Rn	If Rn ≥ Rm with unsigned data, 1 → T	0011nnnnmmmm0010	1	Comparison result
CMP/GE	Rm,Rn	If Rn ≥ Rm with signed data, 1 → T	0011nnnnmmmm0011	1	Comparison result
CMP/HI	Rm,Rn	If Rn > Rm with unsigned data, 1 → T	0011nnnnmmmm0110	1	Comparison result
CMP/GT	Rm,Rn	If Rn > Rm with signed data, 1 → T	0011nnnnmmmm0111	1	Comparison result
CMP/PZ	Rn	If Rn ≥ 0, 1 → T	0100nnnn00010001	1	Comparison result
CMP/PL	Rn	If Rn > 0, 1 → T	0100nnnn00010101	1	Comparison result
CMP/STR	Rm,Rn	If Rn and Rm have an equivalent byte, 1 → T	0010nnnnmmmm1100	1	Comparison result
DIV1	Rm,Rn	Single-step division (Rn/Rm)	0011nnnnmmmm0100	1	Calculation result
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M^ Q → T	0010nnnnmmmm0111	1	Calculation result
DIV0U		0 → M/Q/T	0000000000011001	1	0
DMULS.L	Rm,Rn	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	0011nnnnmmmm1101	2 to 5*	—

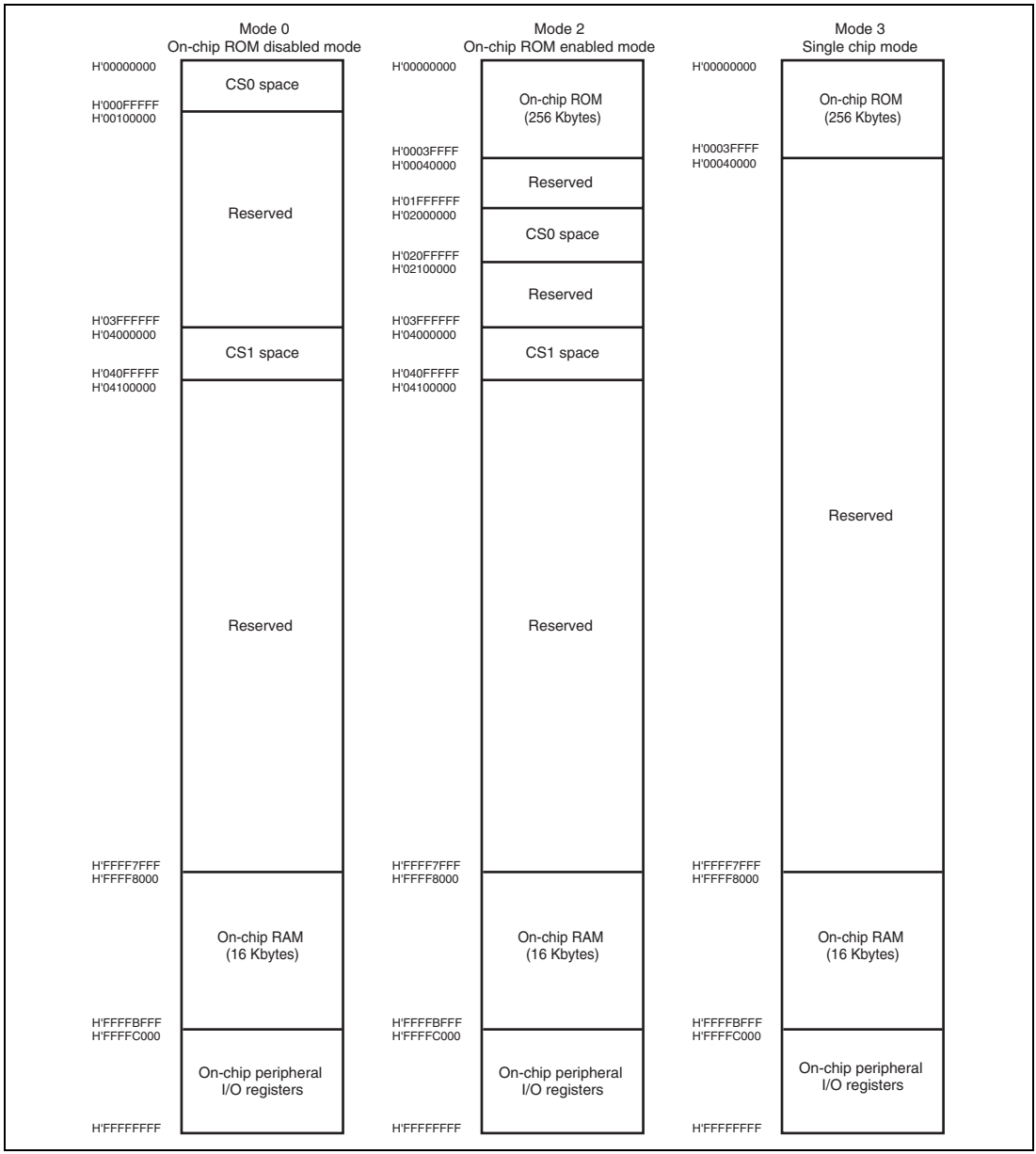


Figure 3.4 Address Map for Each Operating Mode in SH7132 and SH7137 (256-Kbyte Flash Memory Version)

5.1.3 Exception Handling Vector Table

Before exception handling starts, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception handling routines. (The reset exception handling table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets. The vector table addresses are calculated from these vector numbers and vector table address offsets. During exception handling, the start addresses of the exception handling routines are fetched from the exception handling vector table that is indicated by this vector table address.

Table 5.3 shows the vector numbers and vector table address offsets. Table 5.4 shows how vector table addresses are calculated.

Table 5.3 Vector Numbers and Vector Table Address Offsets

Exception Handling Source		Vector Number	Vector Table Address Offset
Power-on reset	PC	0	H'00000000 to H'00000003
	SP	1	H'00000004 to H'00000007
Manual reset	PC	2	H'00000008 to H'0000000B
	SP	3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved for system use)		5	H'00000014 to H'00000017
Illegal slot instruction		6	H'00000018 to H'0000001B
(Reserved for system use)		7	H'0000001C to H'0000001F
		8	H'00000020 to H'00000023
CPU address error		9	H'00000024 to H'00000027
DTC address error		10	H'00000028 to H'0000002B
Interrupt	NMI	11	H'0000002C to H'0000002F
	User break*[1]	12	H'00000030 to H'00000033
(Reserved for system use)		13	H'00000034 to H'00000037
		:	:
		31	H'0000007C to H'0000007F
Trap instruction (user vector)		32	H'00000080 to H'00000083
		:	:
		63	H'000000FC to H'000000FF

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE*1	Transfer Source	Transfer Destination	Priority
MTU2S_3	TGIA_3S	160	H'680	DTCERC3	Arbitrary*2	Arbitrary*2	High
	TGIB_3S	161	H'684	DTCERC2	Arbitrary*2	Arbitrary*2	
	TGIC_3S	162	H'688	DTCERC1	Arbitrary*2	Arbitrary*2	
	TGID_3S	163	H'68C	DTCERC0	Arbitrary*2	Arbitrary*2	
MTU2S_4	TGIA_4S	168	H'6A0	DTCERD15	Arbitrary*2	Arbitrary*2	
	TGIB_4S	169	H'6A4	DTCERD14	Arbitrary*2	Arbitrary*2	
	TGIC_4S	170	H'6A8	DTCERD13	Arbitrary*2	Arbitrary*2	
	TGID_4S	171	H'6AC	DTCERD12	Arbitrary*2	Arbitrary*2	
	TCIV_4S	172	H'6B0	DTCERD11	Arbitrary*2	Arbitrary*2	
MTU2S_5	TGIU_5S	176	H'6C0	DTCERD10	Arbitrary*2	Arbitrary*2	
	TGIV_5S	177	H'6C4	DTCERD9	Arbitrary*2	Arbitrary*2	
	TGIW_5S	178	H'6C8	DTCERD8	Arbitrary*2	Arbitrary*2	
CMT_0	CMI_0	184	H'6E0	DTCERD7	Arbitrary*2	Arbitrary*2	
CMT_1	CMI_1	188	H'6F0	DTCERD6	Arbitrary*2	Arbitrary*2	
A/D_0	ADI_3	208	H'740	DTCERD2	ADDR0 to ADDR7	Arbitrary*2	
A/D_1	ADI_4	212	H'750	DTCERD1	ADDR8 to ADDR15	Arbitrary*2	
SCI_0	RXI_0	217	H'764	DTCERE15	SCRDR_0	Arbitrary*2	
	TXI_0	218	H'768	DTCERE14	Arbitrary*2	SCTDR_0	
SCI_1	RXI_1	221	H'774	DTCERE13	SCRDR_1	Arbitrary*2	
	TXI_1	222	H'778	DTCERE12	Arbitrary*2	SCTDR_1	
SCI_2	RXI_2	225	H'784	DTCERE11	SCRDR_2	Arbitrary*2	
	TXI_2	226	H'788	DTCERE10	Arbitrary*2	SCTDR_2	
SSU	SSRXI	233	H'7A4	DTCERE7	SSRDR0 to SSRDR3	Arbitrary*2	
	SSTXI	234	H'7A8	DTCERE6	Arbitrary*2	SSTDR0 to SSTDR3	
I^2C2	IITXI	238	H'7B8	DTCERE5	Arbitrary*2	ICDRT	
	IIRXI	239	H'7BC	DTCERE4	ICDRR	Arbitrary*2	
RCAN-ET_0	RM0_0	242	H'7C8	DTCERE3	CONTROL0H to CONTROL1L*3	Arbitrary*2	Low

Bit	Bit Name	Initial Value	R/W	Description
0	I1AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1A pin in the TGRA_2 input capture conditions. 0: Does not include the TIOC1A pin in the TGRA_2 input capture conditions 1: Includes the TIOC1A pin in the TGRA_2 input capture conditions

3. Buffer Transfer Control Linked with Interrupt Skipping

   In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

   Figure 10.76 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

   Figure 10.77 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BET0 = 0). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period.

   Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 10.78 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

10.7.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write is executed and the compare match signal is also generated.

Figure 10.122 shows the timing in this case.

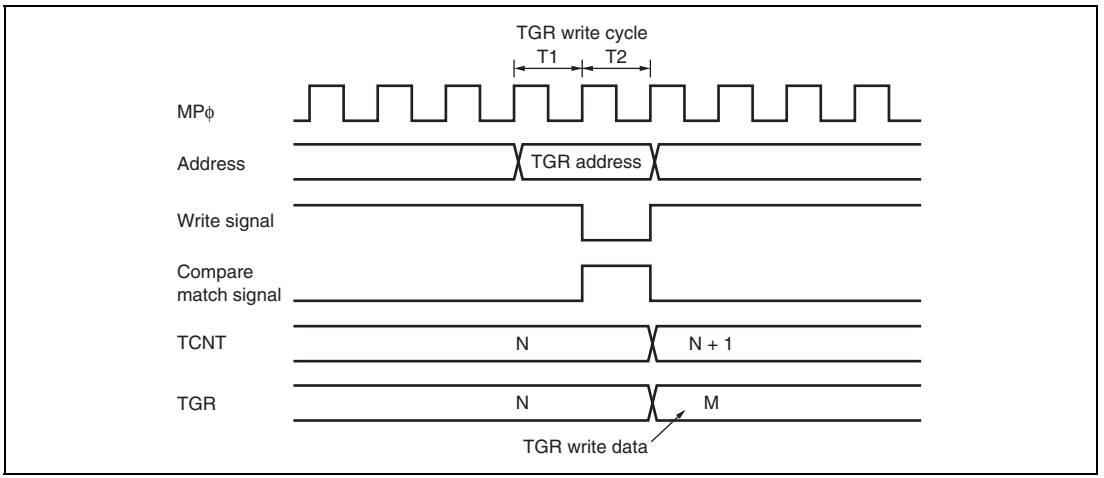
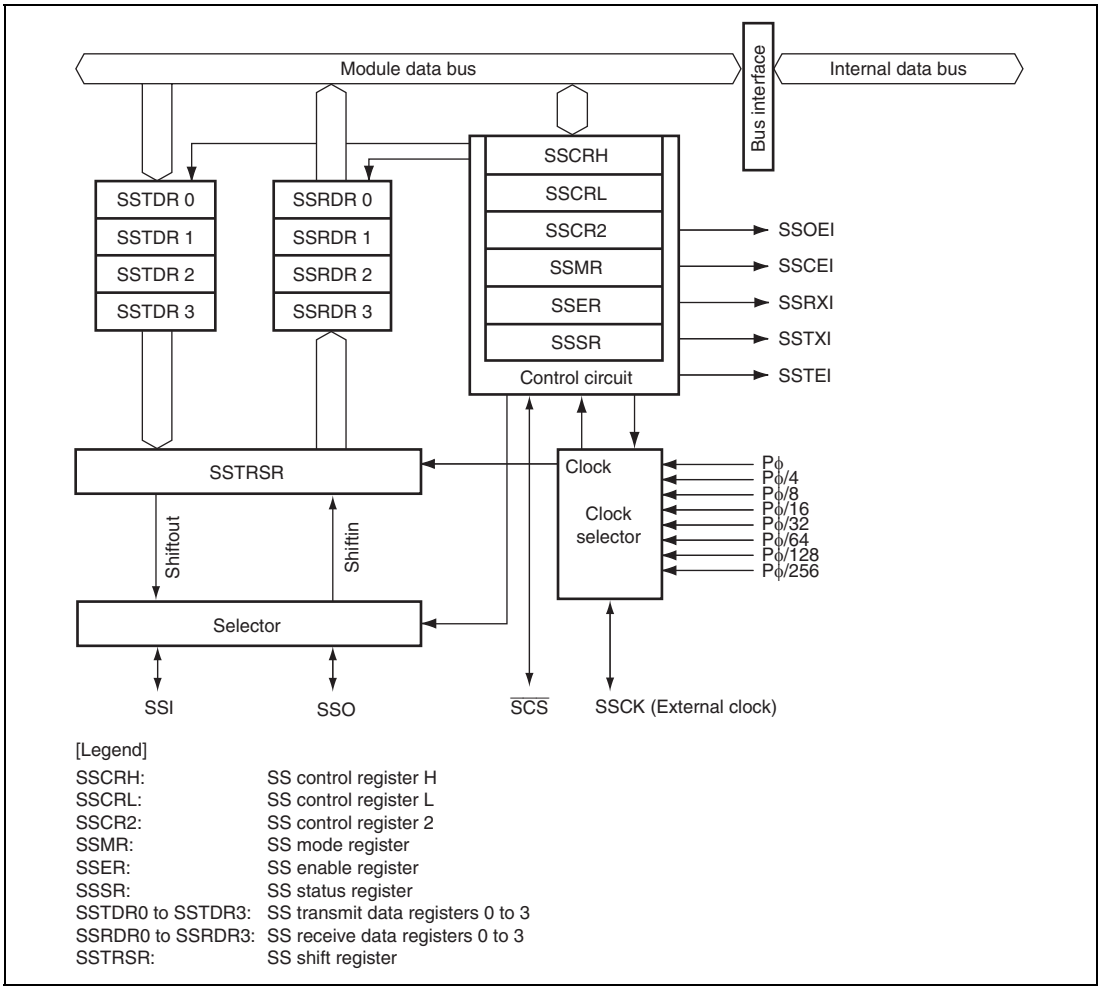


Figure 10.122 Contention between TGR Write and Compare Match

Figure 15.1 shows a block diagram of the SSU.



[Legend]

SSCRH:	SS control register H
SSCRL:	SS control register L
SSCR2:	SS control register 2
SSMR:	SS mode register
SSER:	SS enable register
SSSR:	SS status register
SSTDR0 to SSTDR3:	SS transmit data registers 0 to 3
SSRDR0 to SSRDR3:	SS receive data registers 0 to 3
SSTRSR:	SS shift register

Figure 15.1 Block Diagram of SSU

15.4.4 Communication Modes and Pin Functions

The SSU switches the input/output pin (SSI, SSO, SSCK, and $\overline{SCS}$) functions according to the communication modes and register settings. The input/output directions of the pins should be selected in the port I/O registers. The relationship of communication modes and input/output pin functions are shown in tables 15.5 to 15.7.

Table 15.5 Communication Modes and Pin States of SSI and SSO Pins

Communication Mode	Register Setting					Pin State	
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO
SSU communication mode	0	0	0	0	1	—	Input
				1	0	Output	—
					1	Output	Input
			1	0	1	Input	—
				1	0	—	Output
					1	Input	Output
SSU (bidirectional) communication mode	0	1	0	0	1	—	Input
				1	0	—	Output
			1	0	1	—	Input
				1	0	—	Output
Clock synchronous communication mode	1	0	0	0	1	Input	—
				1	0	—	Output
					1	Input	Output
			1	0	1	Input	—
				1	0	—	Output
					1	Input	Output

[Legend]

—: Not used as SSU pin

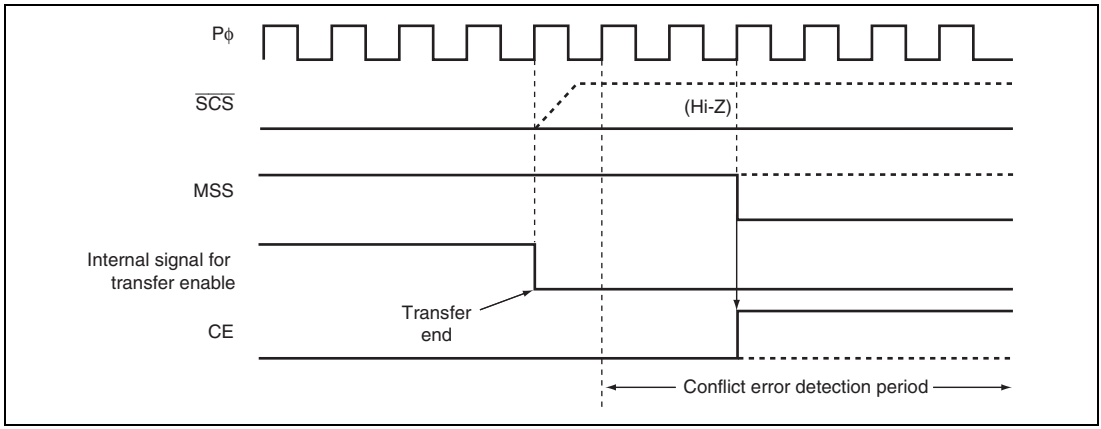


Figure 15.11 Conflict Error Detection Timing (After Transfer End)

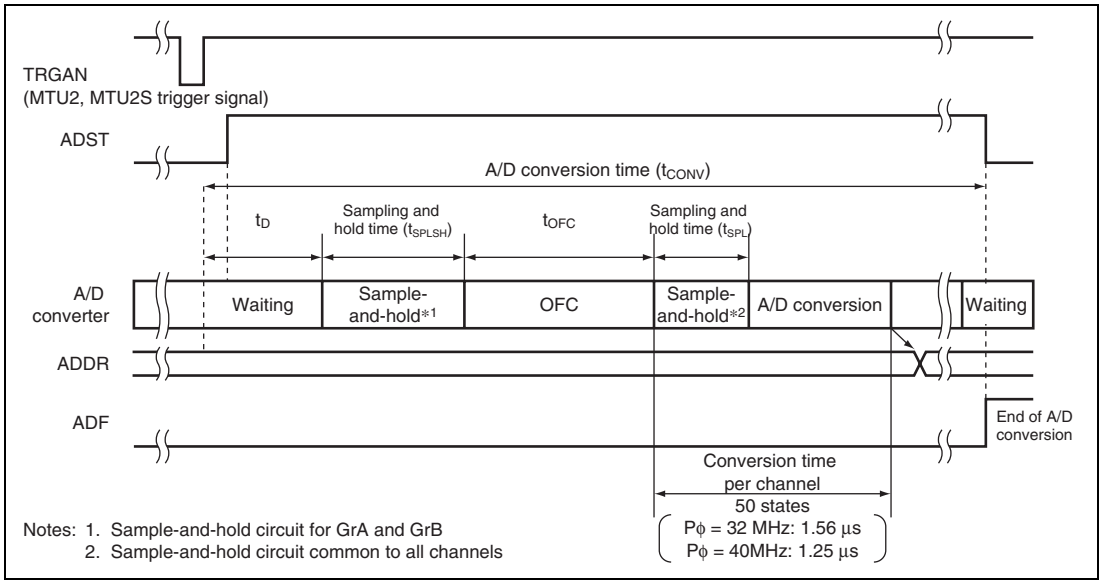


Figure 17.5 A/D Conversion Timing (Single-Cycle Scan Mode)

17.4.4 A/D Converter Activation by MTU2 and MTU2S

A/D conversion is activated by the A/D conversion start triggers (TRGAN, TRG0N, TRG4N, and TRG4BN) from the MTU2 and A/D conversion start triggers (TRGAN, TRG4AN, and TRG4BN) from the MTU2S. To enable this function, set the TRGE bit in ADCR to 1 and clear the EXTRG bit to 0. After this setting is made, if an A/D conversion start trigger from the MTU2 or MTU2S is generated, the ADST bit is set to 1. The timing between the setting of the ADST bit and the start of the A/D conversion is the same for all A/D conversion activation soures.

The A/D conversion start trigger must be input after ADCR, ADSTRGR, and ADANSR registers have been set.

EXT_LAFM[17:0] — Filter mask bits for the CAN Extended identifier [17:0] bits.

EXT_LAFM[17:0]	Description
0	Corresponding EXT_ID bit is cared
1	Corresponding EXT_ID bit is "don't cared"

IDE_LAFM — Filter mask bit for the CAN IDE bit.

IDE_LAFM	Description
0	Corresponding IDE_ID bit is cared
1	Corresponding IDE_ID bit is "don't cared"

(3) Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

Table 20.6 SH7131/SH7136 Multiplexed Pins (Port E)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
E	PE0 I/O (port)	TIOC0A I/O (MTU2)	—	—
	PE1 I/O (port)	TIOC0B I/O (MTU2)	RXD0 input (SCI)	—
	PE2 I/O (port)	TIOC0C I/O (MTU2)	TXD0 output (SCI)	—
	PE3 I/O (port)	TIOC0D I/O (MTU2)	SCK0 I/O (SCI)	—
	PE4 I/O (port)	TIOC1A I/O (MTU2)	RXD1 input (SCI)	—
	PE5 I/O (port)	TIOC1B I/O (MTU2)	TXD1 output (SCI)	—
	PE6 I/O (port)	TIOC2A I/O (MTU2)	SCK1 I/O (SCI)	—
	PE7 I/O (port)	TIOC2B I/O (MTU2)	—	—
	PE8 I/O (port)	TIOC3A I/O (MTU2)	—	—
	PE9 I/O (port)	TIOC3B I/O (MTU2)	—	—
	PE10 I/O (port)	TIOC3C I/O (MTU2)	—	—
	PE11 I/O (port)	TIOC3D I/O (MTU2)	—	—
	PE12 I/O (port)	TIOC4A I/O (MTU2)	—	—
	PE13 I/O (port)	TIOC4B I/O (MTU2)	$\overline{\text{MRES}}$ input (INTC)	—
	PE14 I/O (port)	TIOC4C I/O (MTU2)	—	—
	PE15 I/O (port)	TIOC4D I/O (MTU2)	$\overline{\text{IRQOUT}}$ output (INTC)	—
	PE16 I/O (port)	TIOC3BS I/O (MTU2S)	$\overline{\text{ASEBRKAK}}$ output (E10A)*	$\overline{\text{ASEBRK}}$ input (E10A)*
	PE17 I/O (port)	TIOC3DS I/O (MTU2S)	TCK input (H-UDI)*	—
	PE18 I/O (port)	TIOC4AS I/O (MTU2S)	TDI input (H-UDI)*	—
	PE19 I/O (port)	TIOC4BS I/O (MTU2S)	TDO output (H-UDI)*	—
	PE20 I/O (port)	TIOC4CS I/O (MTU2S)	TMS input (H-UDI)*	—
	PE21 I/O (port)	TIOC4DS I/O (MTU2S)	$\overline{\text{TRST}}$ input (H-UDI)*	—

Note: * Function enabled on the SH7136 only.

Bit	Bit Name	Initial Value	R/W	Description
6	PA9MD2	0	R/W	PA9 Mode
5	PA9MD1	0	R/W	Select the function of the PA9/$\overline{\text{WAIT}}$/TCLKD/TXD2 pin.
4	PA9MD0	0	R/W	000: PA9 I/O (port) 001: TCLKD input (MTU2) 100: $\overline{\text{WAIT}}$ input (BSC)*2 110: TXD2 output (SCI) 111: $\overline{\text{POE8}}$ input (POE) Other than above: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PA8MD2	0*1	R/W	PA8 Mode
1	PA8MD1	0	R/W	Select the function of the PA8/$\overline{\text{WRL}}$/TCLKC/$\overline{\text{POE6}}$/RXD2 pin.
0	PA8MD0	0	R/W	000: PA8 I/O (port) 001: TCLKC input (MTU2) 100: $\overline{\text{WRL}}$ output (BSC)*2 110: RXD2 input (SCI) 111: $\overline{\text{POE6}}$ input (POE) Other than above: Setting prohibited

Notes: 1. The initial value is 1 in the on-chip ROM disabled external-extension mode.
2. This function is available only in the on-chip ROM enabled/disabled external-extension mode. Do not set to this value in single-chip mode.

- PEDRH

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR
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/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PE21DR	0	R/W	See table 21.8.
4	PE20DR	0	R/W	
3	PE19DR	0	R/W	
2	PE18DR	0	R/W	
1	PE17DR	0	R/W	
0	PE16DR	0	R/W	

22.4.3 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, user branch destination address, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. This parameter uses the general registers of the CPU (R4, R5, and R0) or the on-chip RAM area. The initial value is undefined.

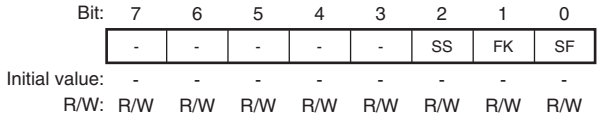
At download all CPU registers are stored, and at initialization or when the on-chip program is executed, CPU registers except for R0 are stored. The return value of the processing result is written in R0. Since the stack area is used for storing the registers or as a work area, the stack area must be saved at the processing start. (The maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameters are used in the following four items.

1. Download control
2. Initialization before programming or erasing
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 22.6.

The processing results of initialization, programming, and erasing are returned, but the bit contents have different meanings according to the processing program. See the description of FPFR for each processing.



Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Unused Return 0.
2	SS	Undefined	R/W	Source Select Error Detect The on-chip program which can be downloaded can be specified as only one type. When more than two types of the program are selected, the program is not selected, or the program is selected without mapping, an error occurs. 0: Download program can be selected normally 1: Download error occurs (Multi-selection or program which is not mapped is selected)
1	FK	Undefined	R/W	Flash Key Register Error Detect Returns the check result whether the value of FKEY is set to H'A5. 0: FKEY setting is normal (FKEY = H'A5) 1: FKEY setting is abnormal (FKEY = value other than H'A5)
0	SF	Undefined	R/W	Success/Fail Returns the result whether download has ended normally or not. 0: Downloading on-chip program has ended normally (no error) 1: Downloading on-chip program has ended abnormally (error occurs)

(2.3) Flash pass/fail result parameter (FPFR: general register R0 of CPU)

This parameter indicates the return value of the initialization result.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	-	-	BR	FQ	SF
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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	Bit Name	Initial Value	R/W	Description
31 to 3	—	Undefined	R/W	Unused Return 0.
2	BR	Undefined	R/W	User Branch Error Detect Returns the check result whether the specified user branch destination address is in the area other than the storage area of the programming/erasing program which has been downloaded. 0: User branch address setting is normal 1: User branch address setting is abnormal
1	FQ	Undefined	R/W	Frequency Error Detect Returns the check result whether the specified operating frequency of the CPU is in the range of the supported operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	Undefined	R/W	Success/Fail Indicates whether initialization is completed normally. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error occurs)

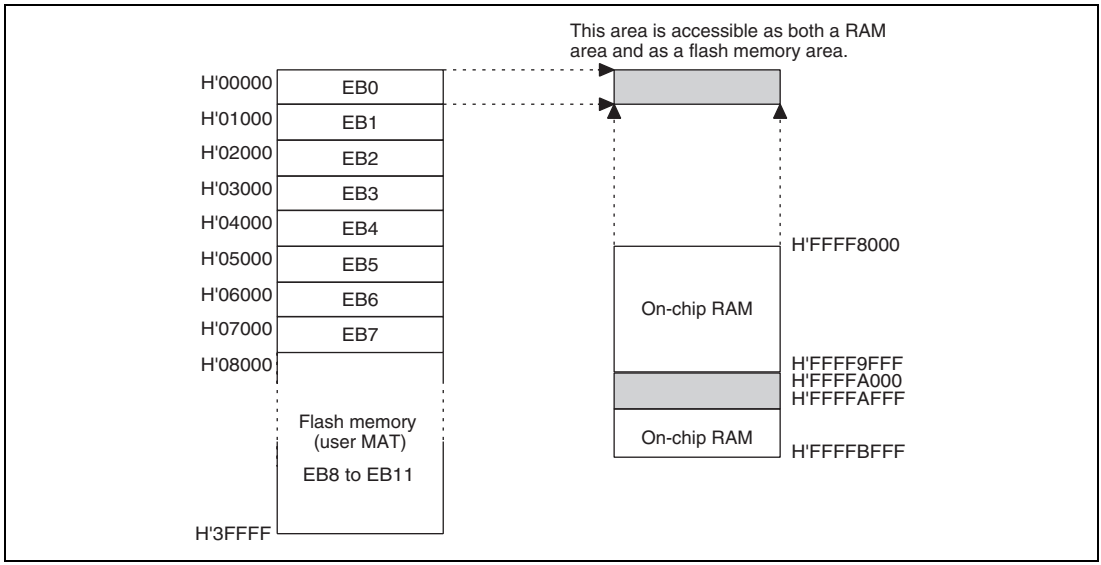


Figure 22.18 Example of Overlapped RAM Operation (256-Kbyte Flash Memory Version)

Figure 22.18 shows an example of an overlap on block area EB0 of the flash memory.

Emulation is possible for a single area selected from among the eight areas, from EB0 to EB7, of the user MAT. The area is selected by the setting of the RAM2 to RAM0 bits in RAMER.

1. To overlap a part of the RAM on area EB0, to allow realtime programming of the data for this area, set the RAMS bit in RAMER to 1, and each of the RAM2 to RAM0 bits to 0.
2. Realtime programming is carried out using the overlaid area of RAM.

In programming or erasing the user MAT, it is necessary to run a program that implements a series of procedural steps, including the downloading of an on-chip program. In this process, set the download area with FTDAR so that the overlaid RAM area and the area where the on-chip program is to be downloaded do not overlap.

Figure 22.19 shows an example of programming data that has been emulated to the EB0 area in the user MAT.

Register Abbreviation	Power-on reset	Manual reset	Software Standby	Deep Software Standby	Module Standby	Sleep	Module
TSR_1	Initialized	Retained	Initialized	Initialized	Initialized	Retained	MTU2
TCNT_1	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TGRA_1	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TGRB_1	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TICCR	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCR_2	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TMDR_2	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TIOR_2	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TIER_2	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TSR_2	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCNT_2	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TGRA_2	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TGRB_2	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCNTU_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TGRU_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCRU_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TIORU_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCNTV_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TGRV_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCRV_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TIORV_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCNTW_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TGRW_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCRW_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TIORW_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TSR_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TIER_5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TSTR5	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCNTCMPCLR	Initialized	Retained	Initialized	Initialized	Initialized	Retained	
TCR_3S	Initialized	Retained	Initialized	Initialized	Initialized	Retained	MTU2S
TCR_4S	Initialized	Retained	Initialized	Initialized	Initialized	Retained	