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	Not For New Designs
Core Processor	H8SX
Core Size	32-Bit Single-Core
Speed	40MHz
Connectivity	CANbus, I²C, SCI, SSU
Peripherals	DMA, Motor Control PWM, PWM, WDT
Number of I/O	95
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	24K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	-
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LFQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df61544j40fpv

Pin No.	Abbreviation in Mode 2, Mode 6, and Mode 7	Abbreviation in Mode 4 and Mode 5
59	P14/SDA1/$\overline{\text{IRQ4}}$-A/$\overline{\text{DREQ1}}$_A	P14/SDA1/$\overline{\text{IRQ4}}$-A/$\overline{\text{DREQ1}}$_A
60	P15/SCL1/$\overline{\text{IRQ5}}$-A/$\overline{\text{TEND1}}$_A	P15/SCL1/$\overline{\text{IRQ5}}$-A/$\overline{\text{TEND1}}$_A
61	P16/SDA0/$\overline{\text{IRQ6}}$-A/$\overline{\text{DACK1}}$_A	P16/SDA0/$\overline{\text{IRQ6}}$-A/$\overline{\text{DACK1}}$_A
62	P17/SCL0/$\overline{\text{IRQ7}}$-A/$\overline{\text{ADTRG1}}$	P17/SCL0/$\overline{\text{IRQ7}}$-A/$\overline{\text{ADTRG1}}$
63	PA0/TDO/PWM0_2	PA0/TDO/PWM0_2
64	PA1/TDI/PWM1_2	PA1/TDI/PWM1_2
65	PA2/TCK/PWM2_2	PA2/TCK/PWM2_2
66	P63/$\overline{\text{IRQ11}}$-B/PWM3_2/TMS/$\overline{\text{DREQ3}}$_B	P63/$\overline{\text{IRQ11}}$-B/PWM3_2/TMS/$\overline{\text{DREQ3}}$_B
67	P64/$\overline{\text{IRQ12}}$-B/CRx_0/$\overline{\text{TEND3}}$_B	P64/$\overline{\text{IRQ12}}$-B/CRx_0/$\overline{\text{TEND3}}$_B
68	P65/$\overline{\text{IRQ13}}$-B/CTx_0/$\overline{\text{DACK3}}$_B	P65/$\overline{\text{IRQ13}}$-B/CTx_0/$\overline{\text{DACK3}}$_B
69	P66/$\overline{\text{IRQ14}}$-B/CRx_1	P66/$\overline{\text{IRQ14}}$-B/CRx_1
70	P67/$\overline{\text{IRQ15}}$-B/CTx_1	P67/$\overline{\text{IRQ15}}$-B/CTx_1
71	Vcc	Vcc
72	Vss	Vss
73	NC	NC
74	P60/TxD4/$\overline{\text{IRQ8}}$-B/$\overline{\text{DREQ2}}$_B	P60/TxD4/$\overline{\text{IRQ8}}$-B/$\overline{\text{DREQ2}}$_B
75	P61/RxD4/$\overline{\text{IRQ9}}$-B/$\overline{\text{TEND2}}$_B	P61/RxD4/$\overline{\text{IRQ9}}$-B/$\overline{\text{TEND2}}$_B
76	P62/SCK4/$\overline{\text{IRQ10}}$-B/$\overline{\text{TRST}}$/$\overline{\text{DACK2}}$_B	P62/SCK4/$\overline{\text{IRQ10}}$-B/$\overline{\text{TRST}}$/$\overline{\text{DACK2}}$_B
77	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$
78	PA3/$\overline{\text{LLWR}}$	PA3/$\overline{\text{LLWR}}$
79	PA4/$\overline{\text{LHWR}}$	PA4/$\overline{\text{LHWR}}$
80	PA5/$\overline{\text{RD}}$	PA5/$\overline{\text{RD}}$
81	PA6/$\overline{\text{AS}}$	PA6/$\overline{\text{AS}}$
82	PA7/Bϕ	PA7/Bϕ
83	P20/TIOCB3/SCK0/SSO_1	P20/TIOCB3/SCK0/SSO_1
84	P21/TIOCA3/RxD0/SSI_1	P21/TIOCA3/RxD0/SSI_1
85	P22/TIOCC3/TxD0/SSCK_1	P22/TIOCC3/TxD0/SSCK_1
86	P23/TIOCD3/$\overline{\text{SCS\_1}}$	P23/TIOCD3/$\overline{\text{SCS\_1}}$
87	P24/TIOCB4	P24/TIOCB4
88	P25/TIOCA4	P25/TIOCA4

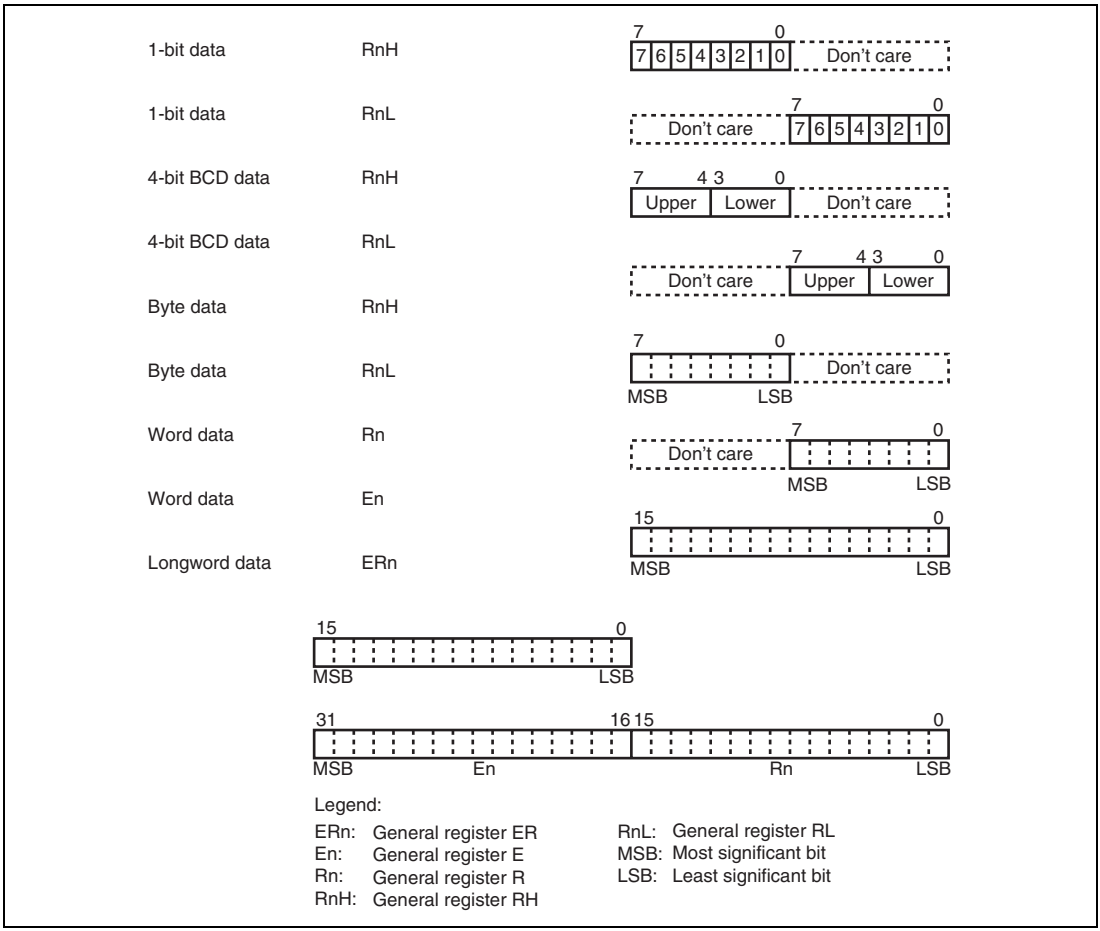


Figure 2.12 General Register Data Formats

2.6.2 Memory Data Formats

Figure 2.13 shows the data formats in memory.

The H8SX CPU can access word data and longword data which are stored at any addresses in memory. When word data begins at an odd address or longword data begins at an address other than a multiple of 4, a bus cycle is divided into two or more accesses. For example, when longword data begins at an odd address, the bus cycle is divided into byte, word, and byte accesses. In this case, these accesses are assumed to be individual bus cycles.

However, instructions to be fetched, word and longword data to be accessed during execution of the stack manipulation, branch table manipulation, block transfer instructions, and MAC instruction should be located to even addresses.

When SP (ER7) is used as an address register to access the stack, the operand size should be word size or longword size.

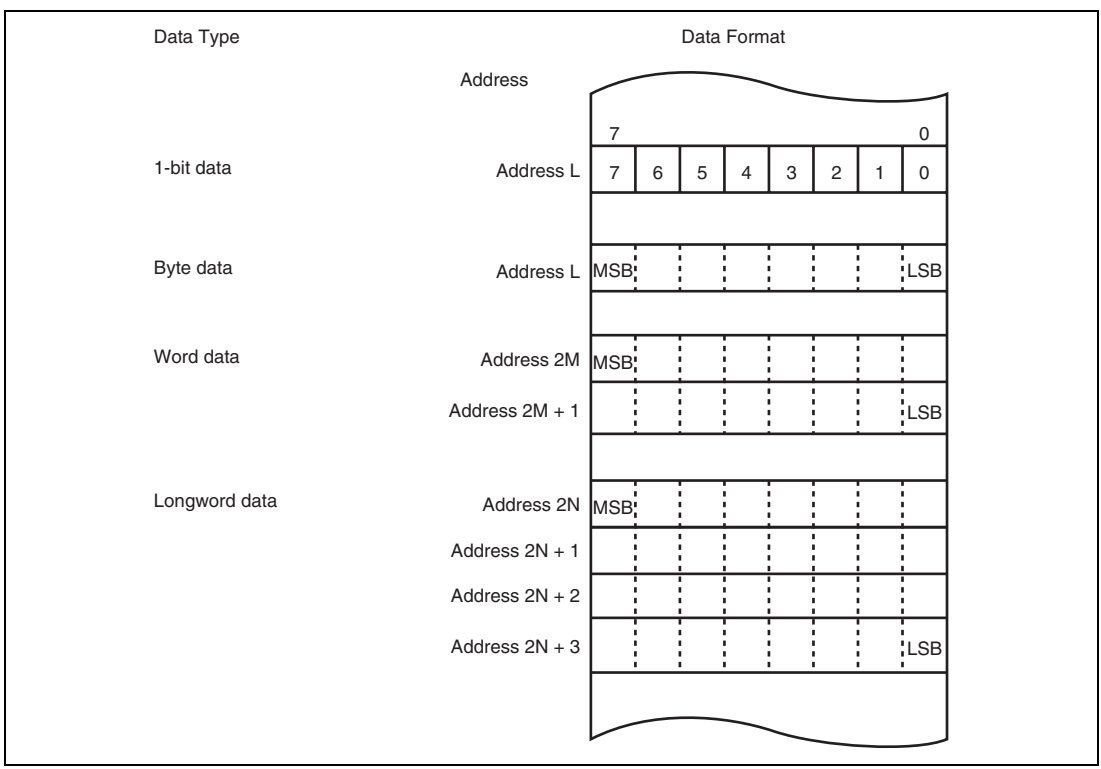
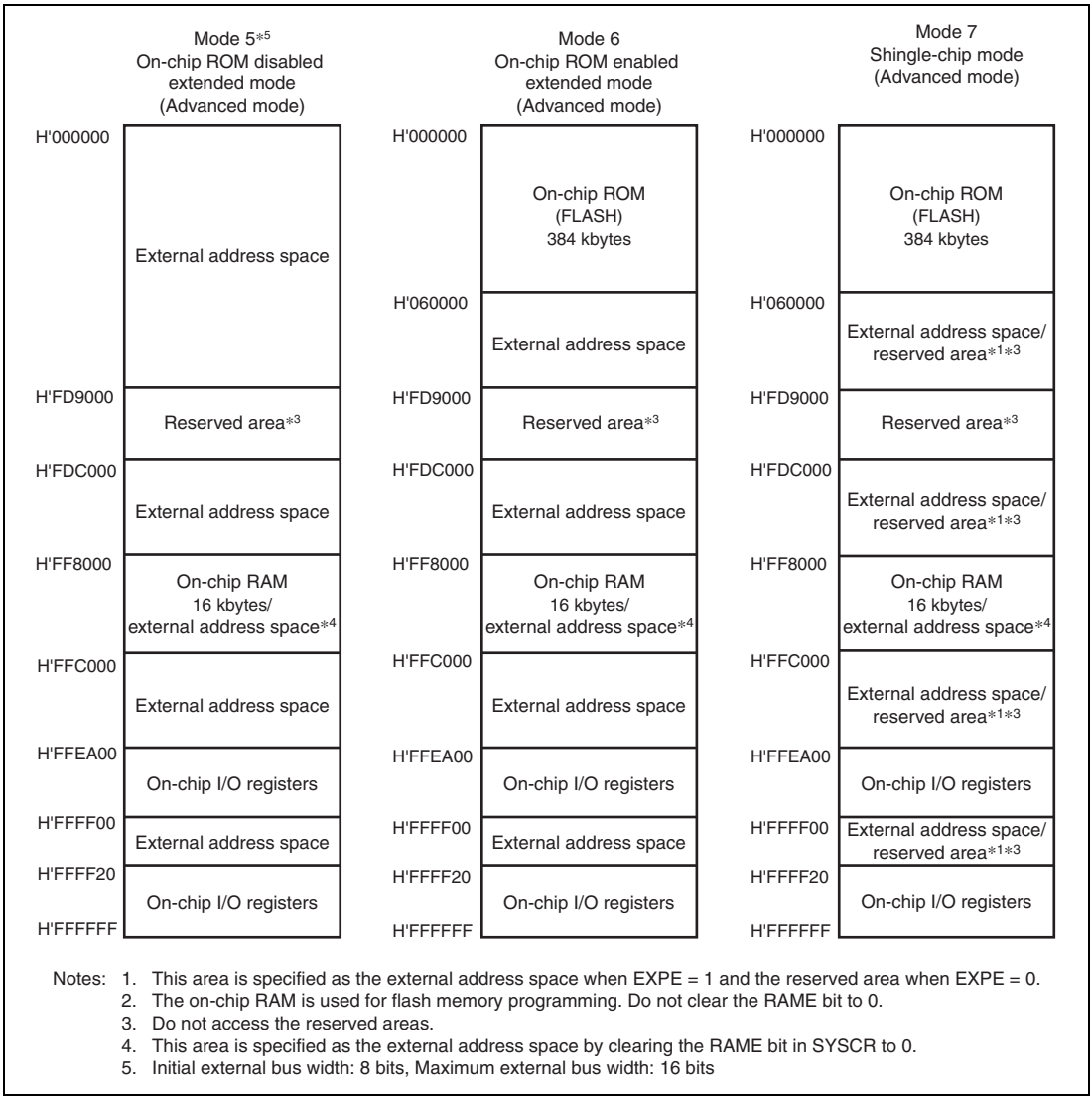


Figure 2.13 Memory Data Formats

Instruction	Size	Function
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder, or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
DIVU	W/L	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: either 16 bits ÷ 16 bits → 16-bit quotient, or 32 bits ÷ 32 bits → 32-bit quotient.
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder, or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
DIVS	W/L	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 16 bits → 16-bit quotient, or 32 bits ÷ 32 bits → 32-bit quotient.
CMP	B/W/L	(EAd) – #IMM, (EAd) – (EAs) Compares data between immediate data, general registers, and memory and stores the result in CCR.
NEG	B/W/L	0 – (EAd) → (EAd) Takes the two's complement (arithmetic complement) of data in a general register or the contents of a memory location.
EXTU	W/L	(EAd) (zero extension) → (EAd) Performs zero-extension on the lower 8 or 16 bits of data in a general register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to longword can be zero-extended.
EXTS	W/L	(EAd) (sign extension) → (EAd) Performs sign-extension on the lower 8 or 16 bits of data in a general register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to longword can be sign-extended.
TAS	B	@ERd – 0, 1 → (<bit 7> of @EAd) Tests memory contents, and sets the most significant bit (bit 7) to 1.
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the result to MAC.
CLRMAC	—	0 → MAC Clears MAC to zero.



Notes: 1. This area is specified as the external address space when EXPE = 1 and the reserved area when EXPE = 0.
2. The on-chip RAM is used for flash memory programming. Do not clear the RAME bit to 0.
3. Do not access the reserved areas.
4. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
5. Initial external bus width: 8 bits, Maximum external bus width: 16 bits

Figure 3.4 H8SX/1543 Address Map (2)

A block diagram of interrupts IRQn is shown in figure 5.2.

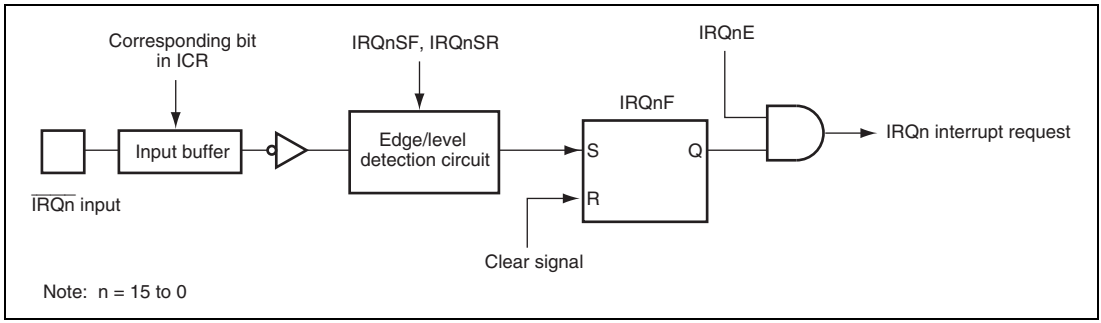


Figure 5.2 Block Diagram of Interrupts IRQn

When ISCR is set so that an IRQn interrupt request is generated at $\overline{\text{IRQn}}$ low-level input, $\overline{\text{IRQn}}$ input should be held low until interrupt handling starts. Then set the corresponding input signal $\overline{\text{IRQn}}$ to high in the interrupt handling routine and clear the IRQnF to 0. Interrupts may not be executed when the corresponding input signal $\overline{\text{IRQn}}$ is set to high before the interrupt handling begins.

5.4.2 Internal Interrupts

The sources for internal interrupts from on-chip peripheral modules have the following features:

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that enable or disable these interrupts. They can be controlled independently. When the enable bit is set to 1, an interrupt request is issued to the interrupt controller.
- The interrupt priority can be set by means of IPR.
- The DMAC can be activated by a TPU, SCI, or other interrupt request.
- The priority level of DMAC activation can be controlled by the DMAC priority control functions.

(7) P21/TIOCA3/RxD0/SSI_1

		Setting		
		SSU*_1	TPU_3	I/O Port
Module Name	Pin Function	SSI_1_OE	TIOCA3_OE	P21DDR
SSU*_1	SSI_1 output	1	—	—
TPU_3	TIOCA3 output	0	1	—
I/O port	P21 output	0	0	1
	P21 input (initial value)	0	0	0

Note: * SSU: Synchronous Serial communication Unit

(8) P20/TIOCB3/SCK0/SSO_1

		Setting			
		SSU*_1	TPU_3	SCI_0	I/O Port
Module Name	Pin Function	SSO_1_OE	TIOCB3_OE	SCK0_OE	P20DDR
SSU*_1	SSO_1 output	1	—	—	—
TPU_3	TIOCB3 output	0	1	—	—
SCI_0	SCK0 output	0	0	1	—
I/O port	P20 output	0	0	0	1
	P20 input (initial value)	0	0	0	0

Note: * SSU: Synchronous Serial communication Unit

(1) Example of PWM Mode Setting Procedure

Figure 9.20 shows an example of the PWM mode setting procedure.

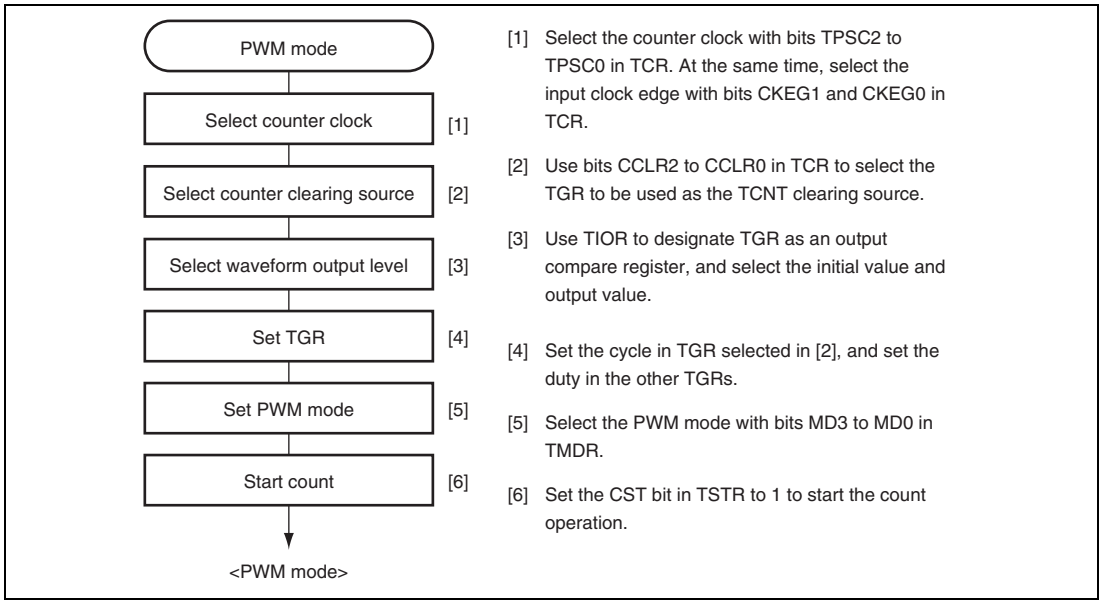


Figure 9.20 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 9.21 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the cycle, and the value set in TGRB register as the duty cycle.

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 9.34 shows the timing when counter clearing by compare match occurrence is specified, and figure 9.35 shows the timing when counter clearing by input capture occurrence is specified.

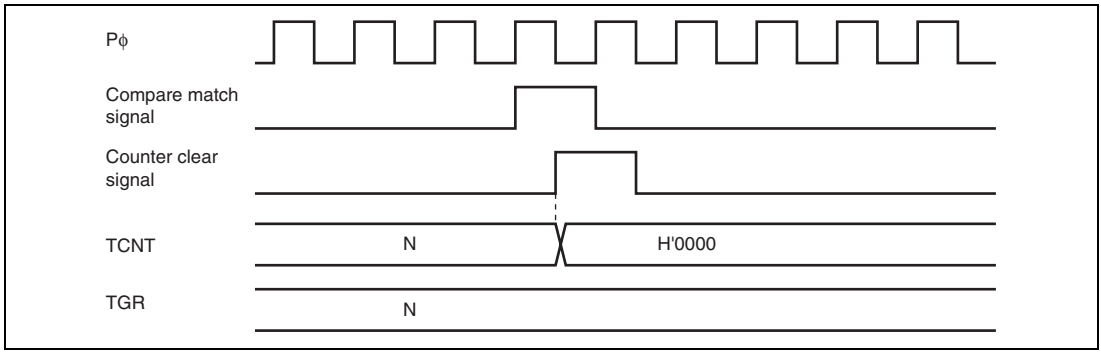


Figure 9.34 Counter Clear Timing (Compare Match)

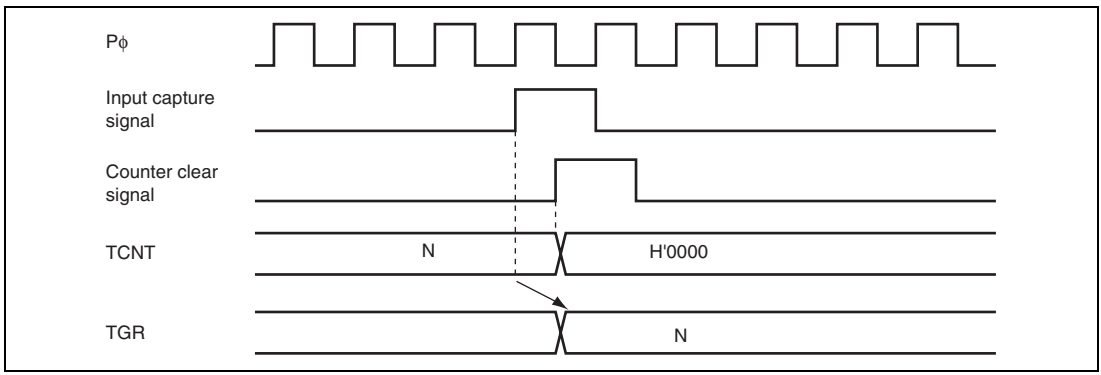


Figure 9.35 Counter Clear Timing (Input Capture)

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: Pϕ clock (n = 0) 01: Pϕ/4 clock (n = 1) 10: Pϕ/16 clock (n = 2) 11: Pϕ/64 clock (n = 3) For the relation between the settings of these bits and the baud rate, see section 12.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 12.3.9, Bit Rate Register (BRR)).

Bit Functions in Smart Card Interface Mode (When SMIF in SCMR = 1):

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode Setting this bit to 1 allows GSM mode operation. In GSM mode, the TEND set timing is put forward to 11.0 etu from the start and the clock output control function is appended. For details, see sections 12.7.6, Data Transmission (Except in Block Transfer Mode) and 12.7.8, Clock Output Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode operation. For details, see section 12.7.3, Block Transfer Mode.
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. Set this bit to 1 in smart card interface mode.
4	O/$\overline{E}$	0	R/W	Parity Mode (valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity For details on the usage of this bit in smart card interface mode, see section 12.7.2, Data Format (Except in Block Transfer Mode).

12.4.1 Data Transfer Format

Table 12.10 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 12.5, Multiprocessor Communication Function.

Table 12.10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transmit/Receive Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STOP
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

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

Bit	Bit Name	Initial Value	R/W	Description
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When receive data is transferred from ICDRS to ICDRR [Clearing conditions] • When 0 is written to this bit after reading RDRF = 1 • When data is read from ICDRR (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is set to 1 [Clearing condition] • When 0 is written to this bit after reading NACKF = 1 (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)
3	STOP	0	R/W	Stop Condition Detection Flag [Setting conditions] • When a stop condition is detected after the frame transfer completion in master mode • In slave mode, when the slave address in the first byte after detecting the start condition and the address set in SAR match, and then a stop condition is detected. [Clearing condition] • When 0 is written to this bit after reading STOP = 1 (When the CPU is used to clear this flag by writing 0 while the corresponding interrupt is enabled, be sure to read the flag after writing 0 to it.)

13.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects a space in the I²C bus shift register, it transfers the transmit data which has been written to ICDRT to ICDRS and starts transmitting data. If the next data is written to ICDRT during transmitting data to ICDRS, continuous transmission is possible.

Bit	7	6	5	4	3	2	1	0
Bit Name								
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

13.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit read-only register that stores the receive data. When one byte of data has been received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register; therefore, this register cannot be written to by the CPU.

Bit	7	6	5	4	3	2	1	0
Bit Name								
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

13.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is an 8-bit write-only register that is used to transmit/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after one by of data is received. This register cannot be read from the CPU.

Bit	7	6	5	4	3	2	1	0
Bit Name								
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

14.10.5 Interrupts

As shown in table 14.10, the Mailbox 0 receive interrupt enables the DMAC activation. When an interrupt is specified as to be activated by the Mailbox 0 receive interrupt and cleared by the interrupt source at the DMA transfer, up to the message control field 1 (CONTROL1) of Mailbox 0 should be read using the block transfer mode.

When clearing the interrupt source flags shown in table 14.10 by the CPU, be sure to read the flag after clearing it in the interrupt service routine. This must be done to prevent the CPU from executing the RTE instruction before the interrupt is cleared in the interrupt controller after the interrupt source flag has been cleared.

15.3.7 SS Transmit Data Registers 0 to 3 (SSTDR0 to SSTDR3)

SSTDR is an 8-bit register that stores transmit data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSTDR0 is valid. When 16-bit data length is selected, SSTDR0 and SSTDR1 are valid. When 24-bit data length is selected, SSTDR0 to SSTDR2 are valid. When 32-bit data length is selected, SSTDR0 to SSTDR3 are valid. Be sure not to access to invalid SSTDR.

When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data has already been written to SSTDR during serial transmission, the SSU performs consecutive serial transmission.

Although SSTDR can always be read from or written to by the CPU and DMAC, to achieve reliable serial transmission, write transmit data to SSTDR after confirming that the TDRE bit in SSSR is set to 1.

• SSTDR0

Bit	7	6	5	4	3	2	1	0
Bit Name								
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

• SSTDR1

Bit	7	6	5	4	3	2	1	0
Bit Name								
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

• SSTDR2

Bit	7	6	5	4	3	2	1	0
Bit Name								
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

• SSTDR3

Bit	7	6	5	4	3	2	1	0
Bit Name								
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

These boot program states are shown in figure 22.18.

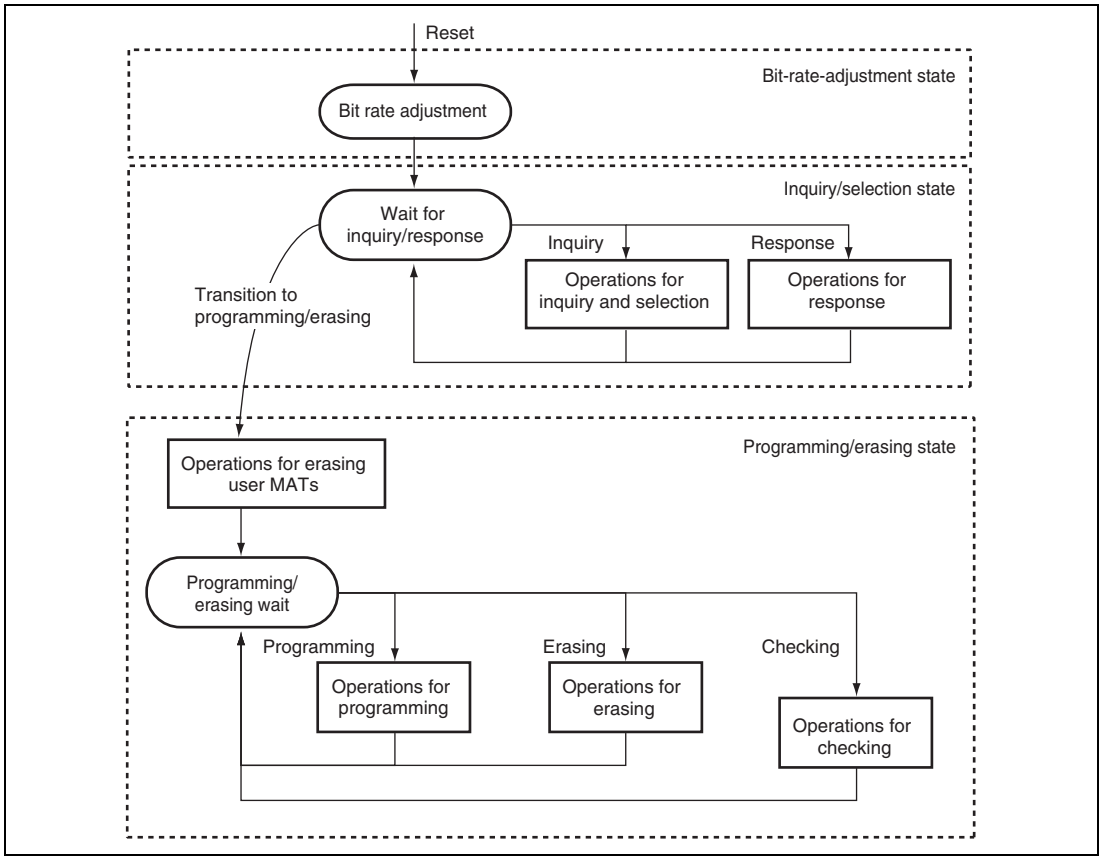


Figure 22.18 Boot Program States

B. Product Lineup

Product Classification	Product Model	Marking	Package (Package Code)
H8SX/1544	R5F61544	R5F61544	LQFP-144 (FP-144L)
H8SX/1543	R5F61543	R5F61543	LQFP-144 (FP-144L)

C. Package Dimensions

For the package dimensions, data in the Renesas IC Package General Catalog has priority.

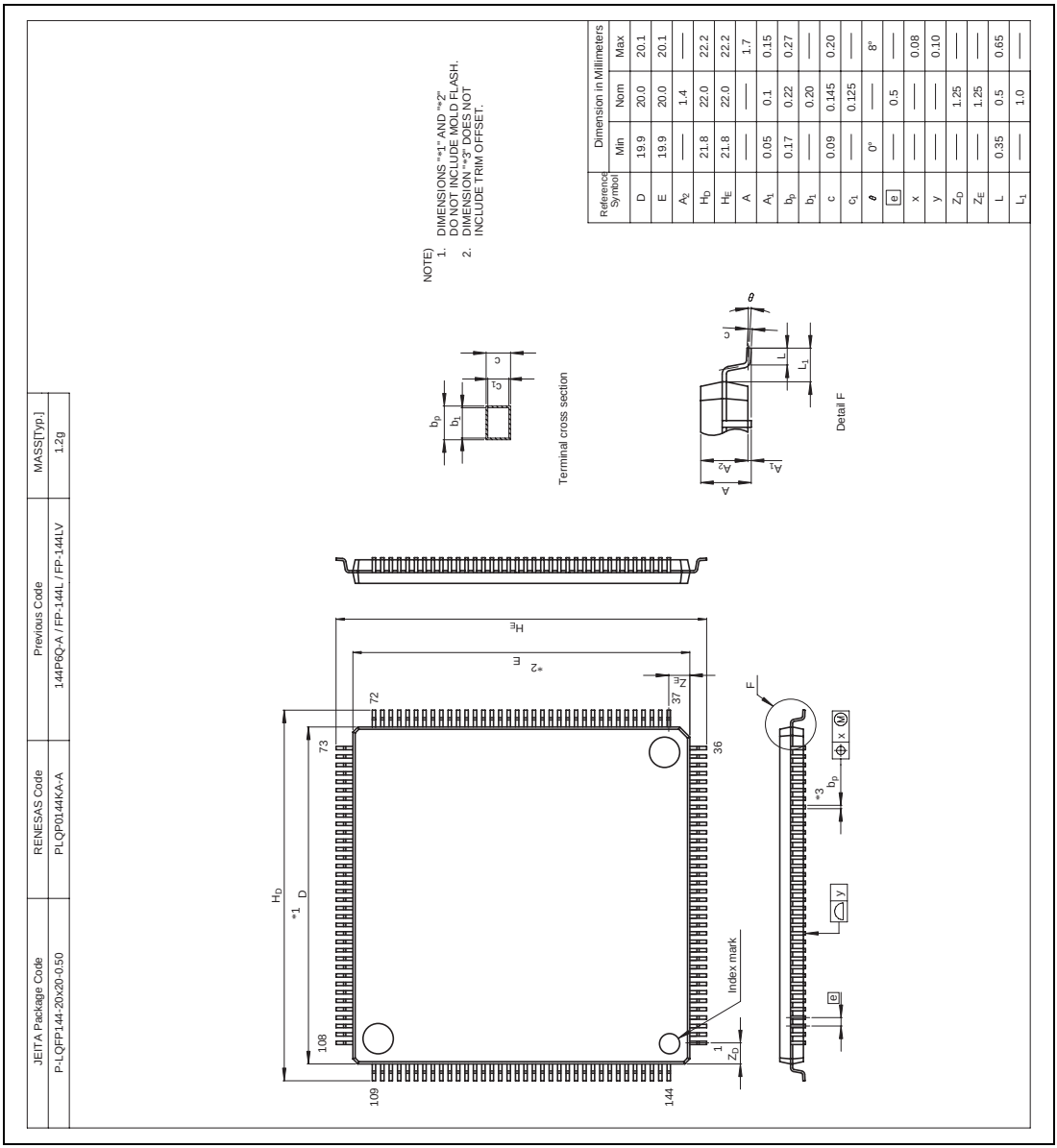


Figure C.1 Package Dimensions (FP-144L)

Q

R