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	Obsolete
Core Processor	SH-2
Core Size	32-Bit Single-Core
Speed	40MHz
Connectivity	CANbus, SCI
Peripherals	POR, PWM, WDT
Number of I/O	53
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	100-BFQFP
Supplier Device Package	100-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/hd64f7047fj40v

User's Manual

32

RENESAS

The revision list can be viewed directly by clicking the title page.

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

SH-2 SH7047 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer SuperH™ RISC engine Family/ SH7000 Series

SH7047F	HD64F7047
SH7049	HD6437049

Renesas Electronics
www.renesas.com

Rev.2.00 2004.09

- On-chip memory

ROM	Model	ROM	RAM	Remarks
Flash memory Version	HD64F7047	256 kbytes	12 kbytes	
Mask ROM Version	HD6437049	128 kbytes	8 kbytes	

- Maximum operating frequency and operating temperature range

Model	Maximum operating frequency (MHz) (system clock (ϕ) and peripheral clock (Pϕ))	Operating temperature range (°C)
HD64F7047F50/HD6437049F50	(50, 25) or (40, 40)	-20 to +75
HD64F7047FW40/HD6437049FW40	(40, 40)	-40 to +85
HD64F7047FJ40/HD6437049FJ40	(40, 40)	-40 to +85

- I/O ports

Model	No. of I/O Pins	No. of Input-only Pins
HD64F7047/HD6437049	53	16

- Supports various power-down states
- Compact package

Model	Package	(Code)	Body Size	Pin Pitch
HD64F7047/HD6437049	QFP-100	FP-100M	14.0 × 14.0 mm	0.5 mm

Instruction Code Format:

Item	Format	Explanation
Instruction	Described in mnemonic. `OP.Sz SRC,DEST`	OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2
Instruction code	Described in MSB ↔ LSB order	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 . . . 1111: R15 iiii: Immediate data dddd: Displacement
Outline of the Operation	→, ←	Direction of transfer
	(xx)	Memory operand
	M/Q/T	Flag bits in the SR
	&	Logical AND of each bit
	|	Logical OR of each bit
	^	Exclusive OR of each bit
	~	Logical NOT of each bit
	<<n	n-bit left shift
	>>n	n-bit right shift
Execution states	—	Value when no wait states are inserted*1
T bit	—	Value of T bit after instruction is executed. An em-dash (—) in the column means no change.

Notes: 1. Instruction execution states: The execution states shown in the table are minimums. The actual number of states may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) equals to the register used by the next instruction.

2. Depending on the operand size, displacement is scaled by ×1, ×2, or ×4. For details, refer the *SH-1/SH-2/SH-DSP Programming Manual*.

Section 8 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 8.1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1.

8.1 Features

- Transfer possible over any number of channels
- Three transfer modes
  Normal, repeat, and block transfer modes available
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 32-bit address space possible
- Activation by software is possible
- Transfer can be set in byte, word, or longword units
- The interrupt that activated the DTC can be requested to the CPU
- Module standby mode can be set

When the wait is specified by software using WCR1, the wait input $\overline{\text{WAIT}}$ signal from outside is sampled. Figure 9.5 shows the $\overline{\text{WAIT}}$ signal sampling. The $\overline{\text{WAIT}}$ signal is sampled at the clock rise one cycle before the clock rise when the T_w state shifts to the T_2 state. When using external waits, use a WCR1 setting of 1 state or more in case of extending $\overline{\text{CS}}$ assertion, and 2 states or more otherwise.

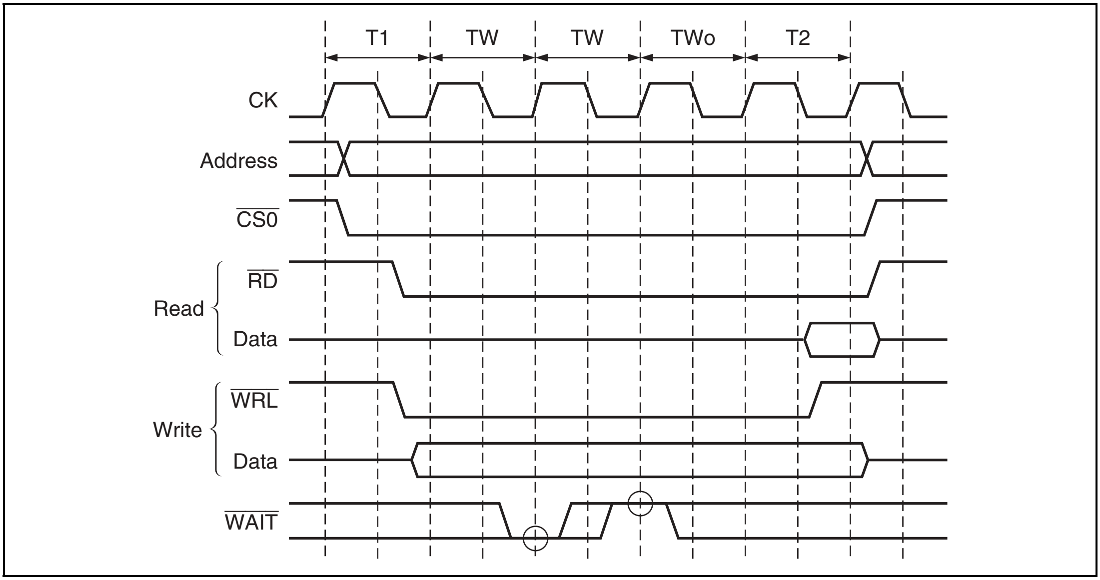


Figure 9.5 Wait State Timing of External Space Access (Two Software Wait States + $\overline{\text{WAIT}}$ Signal Wait State)

- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)
- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register H_4 (TIORH_4)
- Timer I/O control register L_4 (TIORL_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)
- Timer general register C_4 (TGRC_4)
- Timer general register D_4 (TGRD_4)

Common Registers

- Timer start register (TSTR)
- Timer synchro register (TSYR)

Common Registers for timers 3 and 4

- Timer output master enable register (TOER)
- Timer output control enable register (TOCR)
- Timer gate control register (TGCR)
- Timer cycle data register (TCDR)
- Timer dead time data register (TDDR)
- Timer subcounter (TCNTS)
- Timer cycle buffer register (TCBR)

Complementary PWM Mode Output Protection Function: Complementary PWM mode output has the following protection functions.

1. Register and counter miswrite prevention function

   With the exception of the buffer registers, which can be rewritten at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of bit 13 in the bus controller's bus control register 1 (BCR1). Some registers in channels 3 and 4 concerned are listed below: total 21 registers of TCR_3 and TCR_4; TMDR_3 and TMDR_4; TIORH_3 and TIORH_4; TIORL_3 and TIORL_4; TIER_3 and TIER_4; TCNT_3 and TCNT_4; TGRA_3 and TGRA_4; TGRB_3 and TGRB_4; TOER; TOCR; TGCR; TCDR; and TDDR. This function enables the CPU to prevent miswriting due to the CPU runaway by disabling CPU access to the mode registers, control register, and counters. In access disabled state, an undefined value is read from the registers concerned, and cannot be modified.

2. Halting of PWM output by external signal

   The 6-phase PWM output pins can be set automatically to the high-impedance state by inputting specified external signals. There are four external signal input pins.

   See section 10.9, Port Output Enable (POE), for details.

3. Halting of PWM output when oscillator is stopped

   If it is detected that the clock input to this LSI has stopped, the 6-phase PWM output pins automatically go to the high-impedance state. The pin states are not guaranteed when the clock is restarted.

   See section 4.2, Function for Detecting the Oscillator Halt, for details.

10.5 Interrupts

10.5.1 Interrupts and Priorities

There are three kinds of MTU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 6, Interrupt Controller (INTC).

Table 10.42 lists the TPU interrupt sources.

Overflow Interrupt: An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The MTU has five overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The MTU has four underflow interrupts, one each for channels 1 and 2.

10.5.2 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel. For details, see section 8, Data Transfer Controller (DTC).

A total of 17 MTU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1 and 2, and five for channel 4.

10.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match in each channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the MTU conversion start trigger has been selected on the A/D converter at this time, A/D conversion starts.

In the MTU, a total of five TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

12.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Displays whether TDR contains transmit data. [Setting conditions] • Power-on reset, hardware standby mode, or software standby mode • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and transferred data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

14.2 Register Descriptions

The CMT has the following registers for each channel. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Compare Match Timer Start Register (CMSTR)
- Compare Match Timer Control/Status Register_0 (CMCSR_0)
- Compare Match Timer Counter_0 (CMCNT_0)
- Compare Match Timer Constant Register_0 (CMCOR_0)
- Compare Match Timer Control/Status Register_1 (CMCSR_1)
- Compare Match Timer Counter_1 (CMCNT_1)
- Compare Match Timer Constant Register_1 (CMCOR_1)

14.2.1 Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whether to operate or halt the channel 0 and channel 1 counters (CMCNT).

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STR1	0	R/W	Count Start 1 This bit selects whether to operate or halt compare match timer counter_1. 0: CMCNT_1 count operation halted 1: CMCNT_1 count operation
0	STR0	0	R/W	Count Start 0 This bit selects whether to operate or halt compare match timer counter_0. 0: CMCNT_0 count operation halted 1: CMCNT_0 count operation

15.4 Operation

15.4.1 Hardware and Software Resets

The HCAN2 can be reset by hardware or software.

- Hardware Reset

  At power-on reset, manual reset, or in hardware or software standby mode, the HCAN2 is initialized by automatically setting the reset request bit (MCR0) in MCR and the reset status bit (GSR3) in GSR. At the same time, all internal registers, except for mailboxes (MB0 to MB31), are initialized by a hardware reset. Figure 15.5 shows a flowchart in a hardware reset.
- Software Reset

  In the normal operating state, the HCAN2 can be reset by setting the reset request bit (MCR0) in MCR (software reset). In a software reset, if the CAN controller is performing a communication operation (transmission or reception), the HCAN2 enters the initialization state after message transmission or reception has completed. A software reset is enabled after the HCAN2 has entered from the bus off state to the error active state. The reset status bit (GSR3) in GSR is set during initialization. In this initialization, error counters (TEC and REC) are initialized, but other registers and RAM are not initialized.

  Figure 15.6 shows a flowchart in a software reset.

15.4.2 Initialization after Hardware Reset

After a hardware reset, the following initialization processing should be carried out:

1. Clearing of IRR0 bit in the interrupt request register (IRR)
2. Port settings of HCAN2 pins
3. Bit rate setting
4. Mailbox (RAM) initialization
5. Mailbox transmit/receive settings
6. Message transmission method setting

These initial settings must be made while the HCAN2 is in configuration mode. Configuration mode is a state in which the GSR3 bit in GSR is set by a reset. If the MCR0 bit in MCR is cleared to 0, for a while, configuration mode is aborted shortly after the HCAN2 automatically clears the GSR3 bit in GSR. There is a delay between clearing the MCR0 bit and clearing the GSR3 bit because the HCAN2 needs time to be internally reset. After the HCAN2 exits configuration mode, the power-up sequence begins, and communication with the CAN bus is possible as soon as 11 consecutive recessive bits have been detected.

15.5 Interrupt Sources

Table 15.6 lists the HCAN2 interrupt sources. With the exception of the reset processing interrupt (IRR0) by a power-on reset, these sources can be masked. Masking is implemented using the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, refer to section 6, Interrupt Controller (INTC).

Table 15.6 HCAN2 Interrupt Sources

Name	Description	Interrupt Flag	DTC Activation
ERS1	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
	Bus off interrupt (TEC ≥ 256)/bus off recovery interrupt	IRR6	
	Error warning interrupt (TEC ≥ 96)	IRR3	
	Error warning interrupt (REC ≥ 96)	IRR4	
OVR1	Reset processing interrupt by power-on reset	IRR0	Not possible
	Overload frame transmission interrupt	IRR7	
	Unread message overwrite/overrun	IRR9	
	Detection of CAN bus operation in HCAN2 sleep mode	IRR12	
	Timer overflow	IRR13	
	Compare-match condition occurred in TCMR0	IRR14	
	Compare-match condition occurred in TCMR1	IRR15	
RM1	Data frame reception	IRR1	Possible
	Remote frame reception	IRR2	
SLE1	Mailbox empty	IRR8	Not possible

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0 and should only be written with 0.
5	POE6M1	0	R/W	POE6 Mode 1 and 0
4	POE6M0	0	R/W	These bits select the input mode of the $\overline{\text{POE6}}$ pin. 00: Request accepted at falling edge of $\overline{\text{POE6}}$ input 01: $\overline{\text{POE6}}$ input is sampled for low level 16 times every $P\phi/8$ clock, and request is accepted when all samples are low level 10: $\overline{\text{POE6}}$ input is sampled for low level 16 times every $P\phi/16$ clock, and request is accepted when all samples are low level 11: $\overline{\text{POE6}}$ input is sampled for low level 16 times every $P\phi/128$ clock, and request is accepted when all samples are low level
3	POE5M1	0	R/W	POE5 Mode 1 and 0
2	POE5M0	0	R/W	These bits select the input mode of the $\overline{\text{POE5}}$ pin. 00: Request accepted at falling edge of $\overline{\text{POE5}}$ input 01: $\overline{\text{POE5}}$ input is sampled for low level 16 times every $P\phi/8$ clock, and request is accepted when all samples are low level 10: $\overline{\text{POE5}}$ input is sampled for low level 16 times every $P\phi/16$ clock, and request is accepted when all samples are low level 11: $\overline{\text{POE5}}$ input is sampled for low level 16 times every $P\phi/128$ clock, and request is accepted when all samples are low level
1	POE4M1	0	R/W	POE4 Mode 1 and 0
0	POE4M0	0	R/W	These bits select the input mode of the $\overline{\text{POE4}}$ pin. 00: Request accepted at falling edge of $\overline{\text{POE4}}$ input 01: $\overline{\text{POE4}}$ input is sampled for low level 16 times every $P\phi/8$ clock, and request is accepted when all samples are low level 10: $\overline{\text{POE4}}$ input is sampled for low level 16 times every $P\phi/16$ clock, and request is accepted when all samples are low level 11: $\overline{\text{POE4}}$ input is sampled for low level 16 times every $P\phi/128$ clock, and request is accepted when all samples are low level

Note: * Only 0 can be written to clear the flag.

16.8.5 Usage Note

1. To set the POE pin as a level-detective pin, a high level signal must be firstly input to the POE pin.
2. To clear bits POE4F, POE5F, and POE6F to 0, read the ICSR2 register. Clear bits, which are read as 1, to 0, and write 1 to the other bits in the register.

18.5 Port F

Port F is an input-only port with the 16 pins shown in figure 18.5.

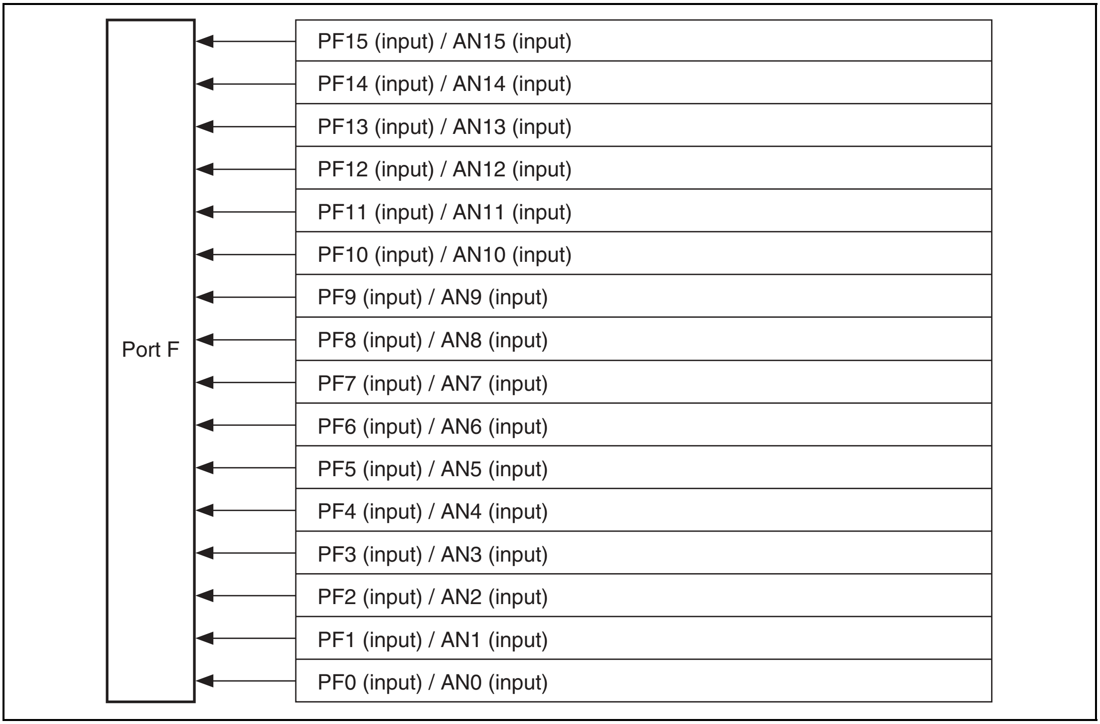


Figure 18.5 Port F

18.5.1 Register Descriptions

Port F is a 16-bit input-only port. Port F has the following register. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

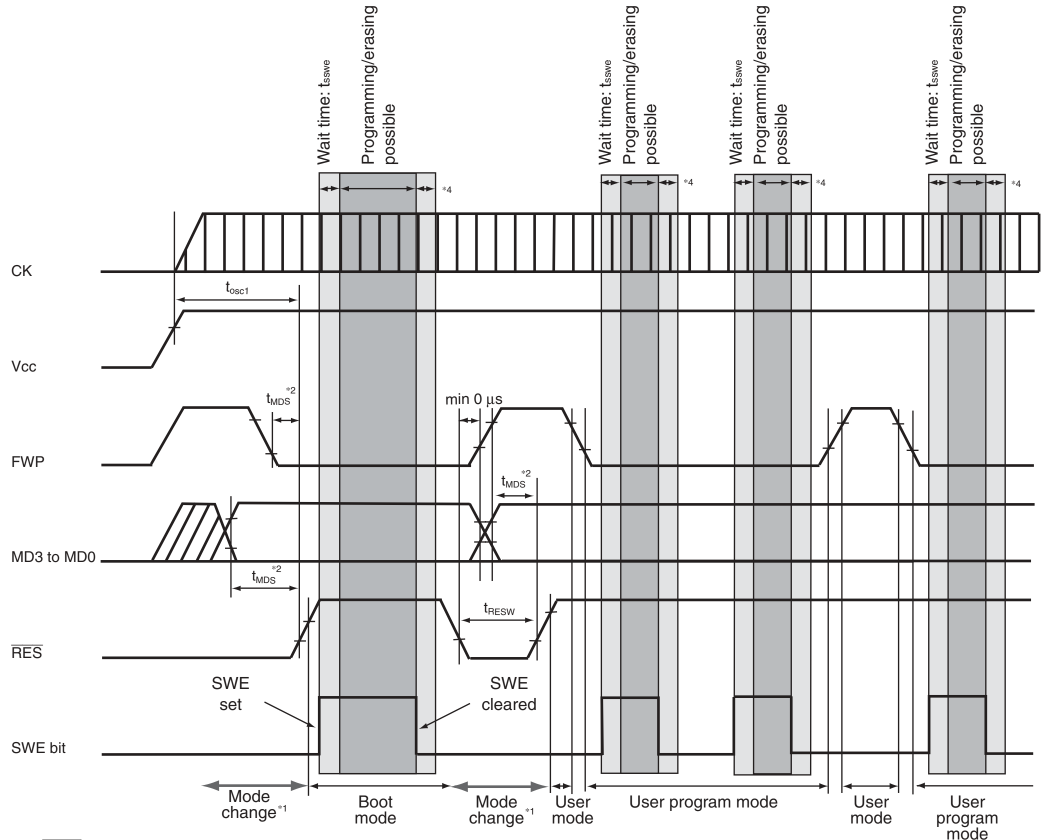
- Port F data register (PFDR)

18.5.2 Port F Data Register (PFDR)

The port F data register (PFDR) is a 16-bit read-only register that stores port F data.

Bits PF15DR to PF0DR correspond to pins PF15 to PF0 (multiplexed functions omitted here).

Any value written into these bits is ignored, and there is no effect on the state of the pins. When any of the bits are read, the pin state rather than the bit value is read directly. However, when an



Period during which flash memory access is prohibited
(t_{sswe}: Wait time after setting SWE bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of RES input.
2. See Section 25.3.3, Control Signal Timing in Electrical Characteristics.
3. See Section 25.5, Flash Memory Characteristics in Electrical Characteristics.
4. Wait time: 100 µs.

Figure 19.13 Mode Transition Timing
(Example: Boot Mode → User Mode →User Program Mode)

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 28[13]	MB28[13]	8	H'FFFFB48E	HCAN2	8, 16	In φ cycles B: 8 W: 8
Mailbox 28[14]	MB28[14]	8	H'FFFFB48F		8	
Mailbox 28[15]	MB28[15]	8	H'FFFFB490		16	
Mailbox 28[16]	MB28[16]	8	H'FFFFB491			
Mailbox 28[17]	MB28[17]	8	H'FFFFB492		16	
Mailbox 28[18]	MB28[18]	8	H'FFFFB493			
Mailbox 29[0]	MB29[0]	8	H'FFFFB4A0		16	
Mailbox 29[1]	MB29[1]	8	H'FFFFB4A1			
Mailbox 29[2]	MB29[2]	8	H'FFFFB4A2		16	
Mailbox 29[3]	MB29[3]	8	H'FFFFB4A3			
Mailbox 29[4]	MB29[4]	8	H'FFFFB4A4		8, 16	
Mailbox 29[5]	MB29[5]	8	H'FFFFB4A5		8	
Mailbox 29[6]	MB29[6]	16	H'FFFFB4A6		16	
Mailbox 29[7]	MB29[7]	8	H'FFFFB4A8		8, 16	
Mailbox 29[8]	MB29[8]	8	H'FFFFB4A9		8	
Mailbox 29[9]	MB29[9]	8	H'FFFFB4AA		8, 16	
Mailbox 29[10]	MB29[10]	8	H'FFFFB4AB		8	
Mailbox 29[11]	MB29[11]	8	H'FFFFB4AC		8, 16	
Mailbox 29[12]	MB29[12]	8	H'FFFFB4AD		8	
Mailbox 29[13]	MB29[13]	8	H'FFFFB4AE		8, 16	
Mailbox 29[14]	MB29[14]	8	H'FFFFB4AF		8	
Mailbox 29[15]	MB29[15]	8	H'FFFFB4B0		16	
Mailbox 29[16]	MB29[16]	8	H'FFFFB4B1			
Mailbox 29[17]	MB29[17]	8	H'FFFFB4B2		16	
Mailbox 29[18]	MB29[18]	8	H'FFFFB4B3			
Mailbox 30[0]	MB30[0]	8	H'FFFFB4C0		16	
Mailbox 30[1]	MB30[1]	8	H'FFFFB4C1			
Mailbox 30[2]	MB30[2]	8	H'FFFFB4C2		16	
Mailbox 30[3]	MB30[3]	8	H'FFFFB4C3			
Mailbox 30[4]	MB30[4]	8	H'FFFFB4C4		8, 16	
Mailbox 30[5]	MB30[5]	8	H'FFFFB4C5		8	
Mailbox 30[6]	MB30[6]	16	H'FFFFB4C6		16	
Mailbox 30[7]	MB30[7]	8	H'FFFFB4C8		8, 16	
Mailbox 30[8]	MB30[8]	8	H'FFFFB4C9		8	
Mailbox 30[9]	MB30[9]	8	H'FFFFB4CA		8, 16	
Mailbox 30[10]	MB30[10]	8	H'FFFFB4CB		8	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
TGRA_1	Initialized	Held	Initialized	Initialized	Initialized	Held	MTU (channel 2)
TGRB_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TMDR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIOR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIER_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TSR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCNT_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRA_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRB_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
IPRA	Initialized	Initialized	Initialized	Held	—	Held	INTC
IPRD	Initialized	Initialized	Initialized	Held	—	Held	
IPRE	Initialized	Initialized	Initialized	Held	—	Held	
IPRF	Initialized	Initialized	Initialized	Held	—	Held	
IPRG	Initialized	Initialized	Initialized	Held	—	Held	
IPRH	Initialized	Initialized	Initialized	Held	—	Held	
ICR1	Initialized	Initialized	Initialized	Held	—	Held	
ISR	Initialized	Initialized	Initialized	Held	—	Held	
IPRI	Initialized	Initialized	Initialized	Held	—	Held	
IPRJ	Initialized	Initialized	Initialized	Held	—	Held	
IPRK	Initialized	Initialized	Initialized	Held	—	Held	
ICR2	Initialized	Initialized	Initialized	Held	—	Held	
PADRL	Initialized	Held	Initialized	Held	—	Held	Port A
PAIORL	Initialized	Held	Initialized	Held	—	Held	
PACRL3	Initialized	Held	Initialized	Held	—	Held	
PACRL1	Initialized	Held	Initialized	Held	—	Held	
PACRL2	Initialized	Held	Initialized	Held	—	Held	
PBDR	Initialized	Held	Initialized	Held	—	Held	Port B
PBIOR	Initialized	Held	Initialized	Held	—	Held	
PBCR1	Initialized	Held	Initialized	Held	—	Held	
PBCR2	Initialized	Held	Initialized	Held	—	Held	
PDDRL	Initialized	Held	Initialized	Held	—	Held	Port D
PDIORL	Initialized	Held	Initialized	Held	—	Held	
PDCRL1	Initialized	Held	Initialized	Held	—	Held	
PDCRL2	Initialized	Held	Initialized	Held	—	Held	
PEDRL	Initialized	Held	Initialized	Held	—	Held	Port E

RENESAS