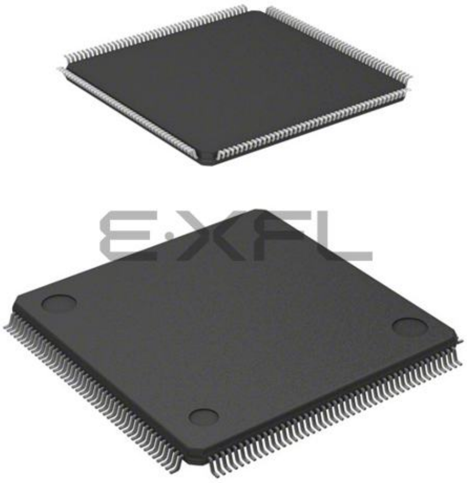
Renesas Electronics America Inc - R5F72145BDFA#V1 Datasheet



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	SH2A-FPU
Core Size	32-Bit Single-Core
Speed	200MHz
Connectivity	CANbus, Ethernet, I²C, SCI, SPI, USB
Peripherals	DMA, PWM, WDT
Number of I/O	112
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	176-LQFP
Supplier Device Package	176-LFQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f72145bdfa-v1

When exception handling starts, the CPU operates as follows:

(1) Exception Handling Triggered by Reset

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 5.1.3, Exception Handling Vector Table, for more information. The vector base register (VBR) is then initialized to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the interrupt controller (INTC) is also initialized to 0. The program begins running from the PC address fetched from the exception handling vector table.

(2) Exception Handling Triggered by Address Errors, Register Bank Errors, Interrupts, and Instructions

SR and PC are saved to the stack indicated by R15. In the case of interrupt exception handling other than NMI or UBC with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved to the register banks. In the case of exception handling due to an address error, register bank error, NMI interrupt, UBC interrupt, or instruction, saving to a register bank is not performed. When saving is performed to all register banks, automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the INTC is 1), register bank overflow exception will be generated. In the case of interrupt exception handling, the interrupt priority level is written to the I3 to I0 bits in SR. In the case of exception handling due to an address error or instruction, the I3 to I0 bits are not affected. The start address is then fetched from the exception handling vector table and the program begins running from that address.

Table 8.10 Number of Cycles Required for Each Execution State

Object to be Accessed		On-Chip RAM*1	Flash Memory (ROM)	On-Chip I/O Registers*4			External Device*5		
Bus width		32 bits	32 bits	8 bits*4	16 bits	32 bits	8 bits	16 bits	32 bits
Access cycles		1Bϕ to 4Bϕ*1*2	3Bϕ to 4Iϕ + 3Bϕ*2	2Pϕ	2Pϕ	2Pϕ	2Bϕ	2Bϕ	2Bϕ
Execution status	Vector read S_I	1Bϕ to 4Bϕ*1*2	3Bϕ to 4Iϕ + 3Bϕ*2	—	—	—	9Bϕ	5Bϕ	3Bϕ
	Transfer information read S_J	1Bϕ to 4Bϕ*1	—	—	—	—	9Bϕ	5Bϕ	3Bϕ
	Transfer information write S_k	1Bϕ to 3Bϕ*1	—	—	—	—	2Bϕ*6	2Bϕ*6	2Bϕ*6
	Byte data read S_L	1Bϕ to 4Bϕ*1	—	1Bϕ + 2Pϕ*3	1Bϕ + 2Pϕ*3	—	3Bϕ	3Bϕ	3Bϕ
	Word data read S_L	1Bϕ to 4Bϕ*1	—	1Bϕ + 2Pϕ*3	1Bϕ + 2Pϕ*3	—	5Bϕ	3Bϕ	3Bϕ
	Longword data read S_L	1Bϕ to 4Bϕ*1	—	1Bϕ + 4Pϕ*3	1Bϕ + 2Pϕ*3	1Bϕ + 4Pϕ*3	9Bϕ	5Bϕ	3Bϕ
	Byte data write S_M	1Bϕ to 3Bϕ*1	—	1Bϕ + 2Pϕ*3	1Bϕ + 2Pϕ*3	—	2Bϕ*6	2Bϕ*6	2Bϕ*6
	Word data write S_M	1Bϕ to 3Bϕ*1	—	1Bϕ + 2Pϕ*3	1Bϕ + 2Pϕ*3	—	2Bϕ*6	2Bϕ*6	2Bϕ*6
	Longword data write S_M	1Bϕ to 3Bϕ*1	—	1Bϕ + 4Pϕ*3	1Bϕ + 2Pϕ*3	1Bϕ + 4Pϕ*3	2Bϕ*6	2Bϕ*6	2Bϕ*6
	Internal operation S_N	1							

Notes: 1. Values for on-chip RAM. Number of cycles varies depending on the ratio of Iϕ:Bϕ.

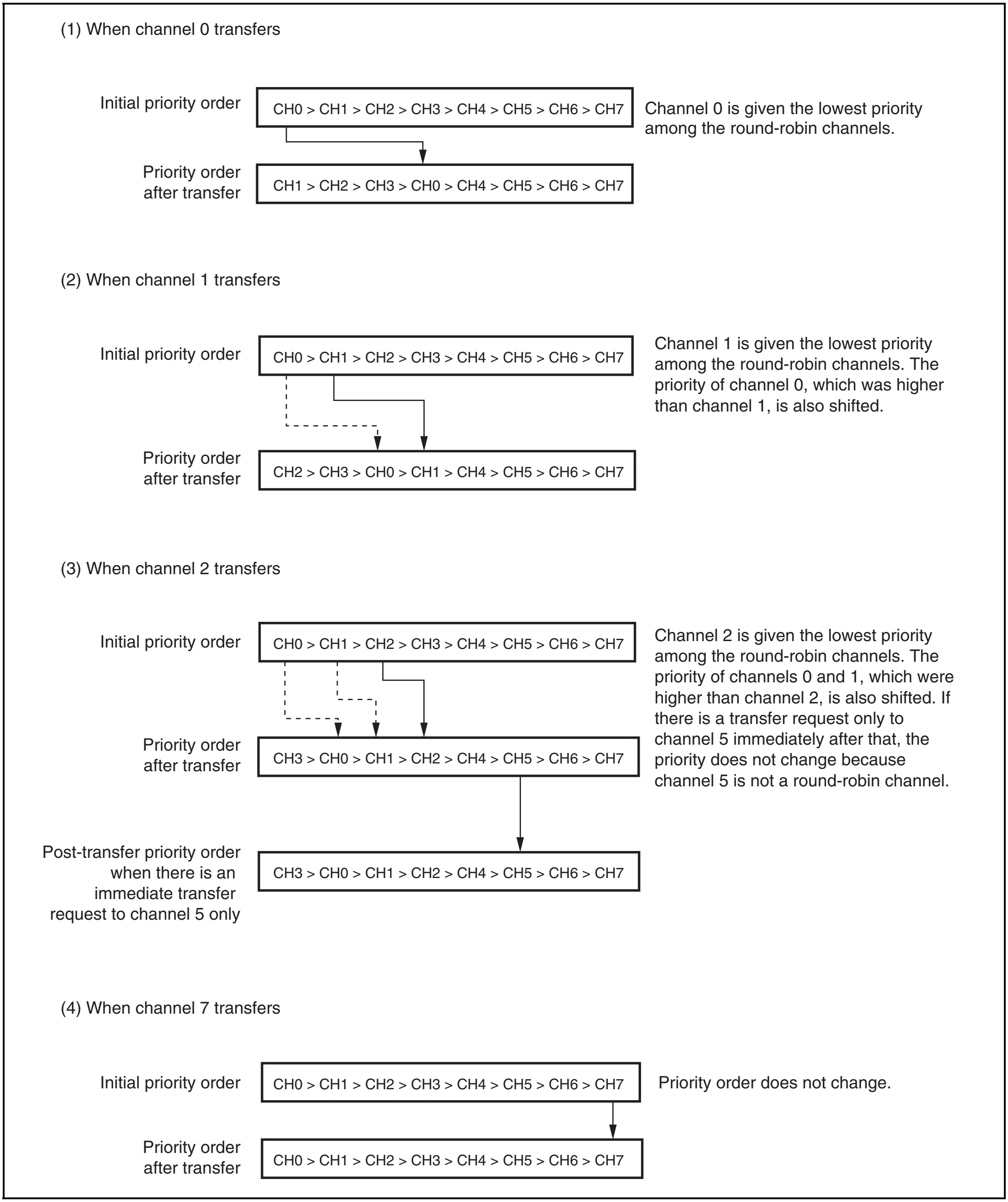
	Read	Write
Iϕ:Bϕ = 1:1	3Bϕ, 4Bϕ	2Bϕ, 3Bϕ
Iϕ:Bϕ = 1:1/2	2Bϕ, 3Bϕ	2Bϕ
Iϕ:Bϕ = 1:1/4	2Bϕ	1Bϕ, 2Bϕ
Iϕ:Bϕ = 1:1/8	1Bϕ	1Bϕ

6. SRAM interface with byte selection
   - Can connect directly to a SRAM with byte selection.
7. Burst ROM interface (clock synchronous)
   - Can connect directly to a ROM of the clock-synchronous type.
8. Bus arbitration
   - Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.
9. Refresh function
   - Supports the auto-refresh and self-refresh functions.
   - Specifies the refresh interval using the refresh counter and clock selection.
   - Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8).
10. Usage as interval timer for refresh counter
    - Generates an interrupt request at compare match.

Figure 9.1 shows a block diagram of the BSC.

Table 9.21 Conditions for Determining Number of Idle Cycles

No.	Condition	Description	Range	Note
(1)	DMAIW[2:0] in CMNCR	These bits specify the number of idle cycles for DMA single address transfer. This condition is effective only for single address transfer and generates idle cycles after the access is completed.	0 to 12	When 0 is specified for the number of idle cycles, the DACK signal may be asserted continuously. This causes a discrepancy between the number of cycles detected by the device with DACK and the DMAC transfer count, resulting in a malfunction.
(2)	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access other than single address transfer. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS1 space followed by reading other CS space, the bits IWRRD[2:0] in CS1BCR should be set to B'100 to specify six or more idle cycles. This condition is effective only for access cycles other than single address transfer and generates idle cycles after the access is completed.	0 to 12	Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively.
(3)	SDRAM-related bits in CSnWCR	These bits specify precharge completion and startup wait cycles and idle cycles between commands for SDRAM access. This condition is effective only for SDRAM access and generates idle cycles after the access is completed	0 to 3	Specify these bits in accordance with the specification of the target SDRAM.
(4)	WM in CSnWCR	This bit enables or disables external $\overline{\text{WAIT}}$ pin input for the memory types other than SDRAM. When this bit is cleared to 0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to 1 (disabled), no idle cycle is generated.	0 or 1	


(1) When channel 0 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
Priority order after transfer
CH1 > CH2 > CH3 > CH0 > CH4 > CH5 > CH6 > CH7
(2) When channel 1 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
Priority order after transfer
CH2 > CH3 > CH0 > CH1 > CH4 > CH5 > CH6 > CH7
(3) When channel 2 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
Priority order after transfer
CH3 > CH0 > CH1 > CH2 > CH4 > CH5 > CH6 > CH7
Post-transfer priority order when there is an immediate transfer request to channel 5 only
CH3 > CH0 > CH1 > CH2 > CH4 > CH5 > CH6 > CH7
(4) When channel 7 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
Priority order after transfer
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7

Channel 0 is given the lowest priority among the round-robin channels.

Channel 1 is given the lowest priority among the round-robin channels. The priority of channel 0, which was higher than channel 1, is also shifted.

Channel 2 is given the lowest priority among the round-robin channels. The priority of channels 0 and 1, which were higher than channel 2, is also shifted. If there is a transfer request only to channel 5 immediately after that, the priority does not change because channel 5 is not a round-robin channel.

Priority order does not change.

Figure 10.3 Round-Robin Mode

Table 11.11 Setting of Operation Mode by Bits MD0 to MD3

Bit 3 MD3	Bit 2 MD2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Setting prohibited
		1	0	PWM mode 1
			1	PWM mode 2*[1]
	1	0	0	Phase counting mode 1*[2]
			1	Phase counting mode 2*[2]
		1	0	Phase counting mode 3*[2]
			1	Phase counting mode 4*[2]
1	0	0	0	Reset synchronous PWM mode*[3]
			1	Setting prohibited
		1	X	Setting prohibited
	1	0	0	Setting prohibited
			1	Complementary PWM mode 1 (transmit at crest)*[3]
		1	0	Complementary PWM mode 2 (transmit at trough)*[3]
			1	Complementary PWM mode 2 (transmit at crest and trough)*[3]

[Legend]

X: Don't care

Notes: 1. PWM mode 2 cannot be set for channels 3 and 4.

2. Phase counting mode cannot be set for channels 0, 3, and 4.

3. Reset synchronous PWM mode, complementary PWM mode can only be set for channel 3. When channel 3 is set to reset synchronous PWM mode or complementary PWM mode, the channel 4 settings become ineffective and automatically conform to the channel 3 settings. However, do not set channel 4 to reset synchronous PWM mode or complementary PWM mode. Reset synchronous PWM mode and complementary PWM mode cannot be set for channels 0, 1, and 2.

Bit	Bit Name	Initial Value	R/W	Description
1	SCH3S	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_3S in the MTU2S. 0: Does not specify synchronous start for TCNT_3S in the MTU2S 1: Specifies synchronous start for TCNT_3S in the MTU2S [Clearing condition] • When 1 is set to the CST3 bit of TSTRS in MTU2S while SCH3S = 1
0	SCH4S	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_4S in the MTU2S. 0: Does not specify synchronous start for TCNT_4S in the MTU2S 1: Specifies synchronous start for TCNT_4S in the MTU2S [Clearing condition] • When 1 is set to the CST4 bit of TSTRS in MTU2S while SCH4S = 1

Note: Only 1 can be written to set the register.

Bit	Bit Name	Initial Value	R/W	Description
0	WRE	0	R/(W)	Initial Output Suppression Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is suppressed only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value is also output when synchronous clearing occurs in the Tb interval at the trough immediately after TCNT_3 and TCNT_4 start operation. For the Tb interval at the trough in complementary PWM mode, see figure 11.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

Note: * Do not set to 1 when complementary PWM mode is not selected.

11.3.33 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCOR), and timer A/D converter start request cycle set buffer registers (TADCOBR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. These are connected to the CPU by a 16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

Figure 11.6 illustrates periodic counter operation.

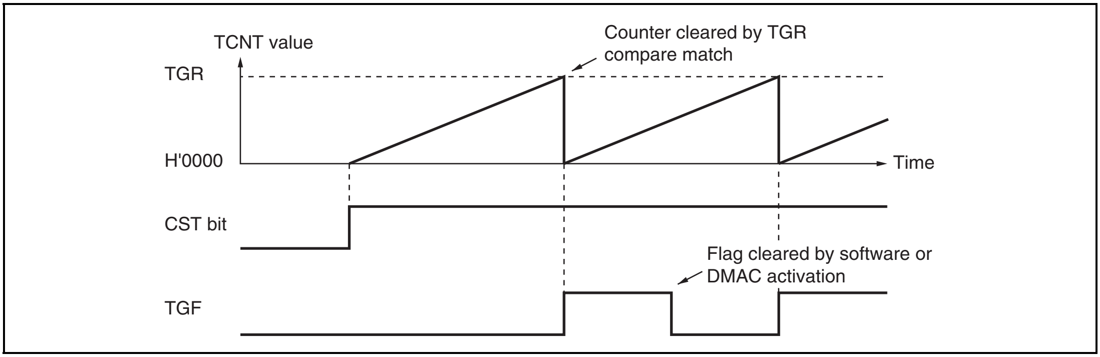


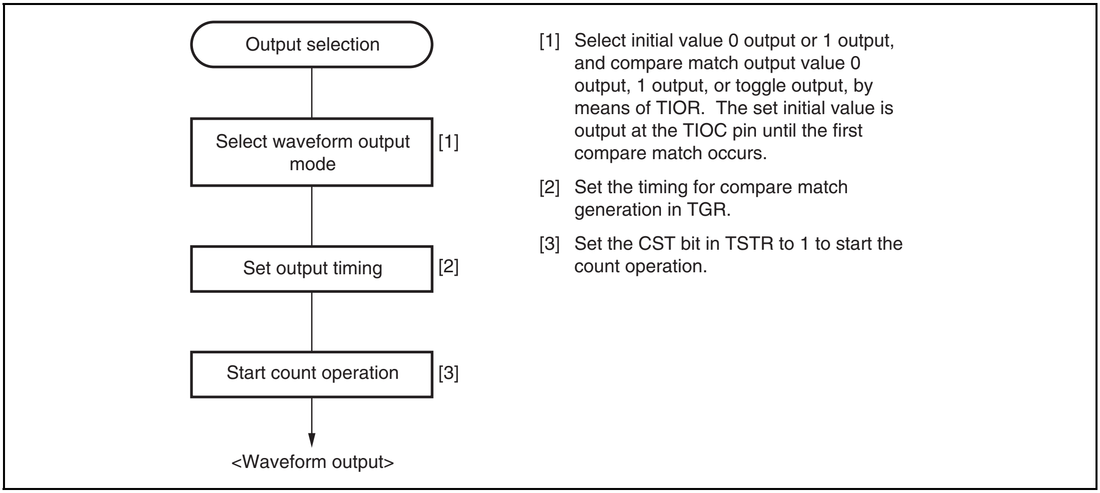
Figure 11.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU2 can perform 0, 1, or toggle output from the corresponding output pin using compare match.

(a) Example of Setting Procedure for Waveform Output by Compare Match

Figure 11.7 shows an example of the setting procedure for waveform output by compare match.



[1] Select initial value 0 output or 1 output, and compare match output value 0 output, 1 output, or toggle output, by means of TIOR. The set initial value is output at the TIOC pin until the first compare match occurs.

[2] Set the timing for compare match generation in TGR.

[3] Set the CST bit in TSTR to 1 to start the count operation.

Figure 11.7 Example of Setting Procedure for Waveform Output by Compare Match

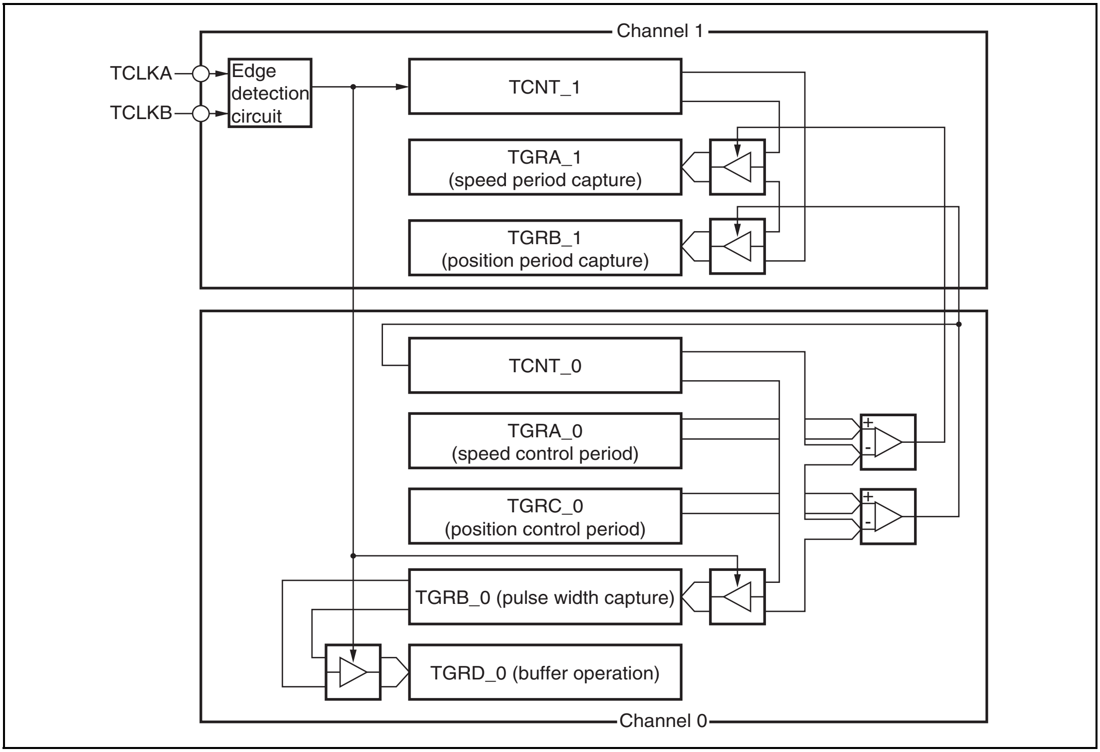

Channel 1
TCLKA
TCLKB
Edge detection circuit
TCNT_1
TGRA_1 (speed period capture)
TGRB_1 (position period capture)
TCNT_0
TGRA_0 (speed control period)
TGRC_0 (position control period)
+
-
+
-
TGRB_0 (pulse width capture)
TGRD_0 (buffer operation)
Channel 0

Figure 11.34 Phase Counting Mode Application Example

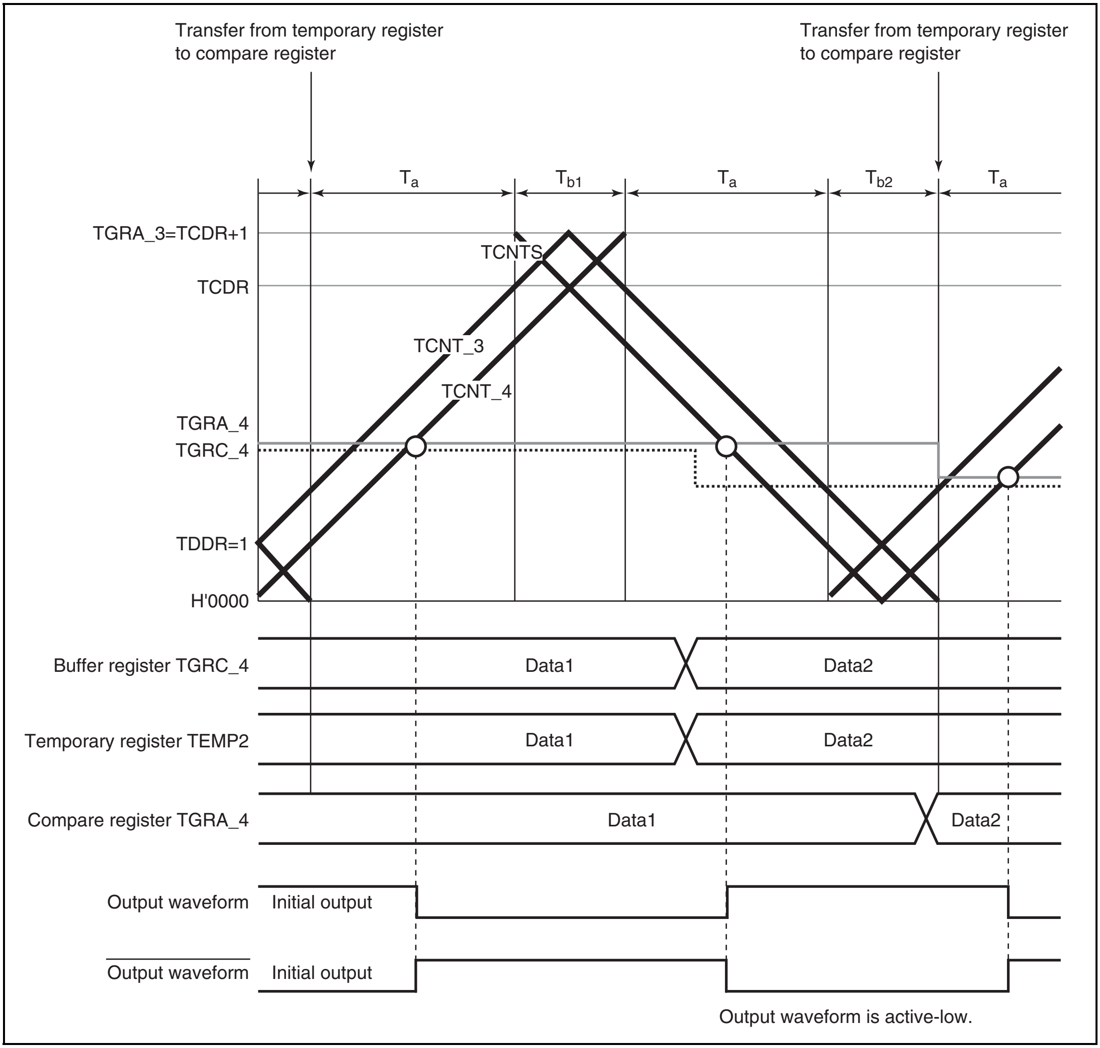


Figure 11.41 Example of Operation without Dead Time

(17) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode

Figure 11.155 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

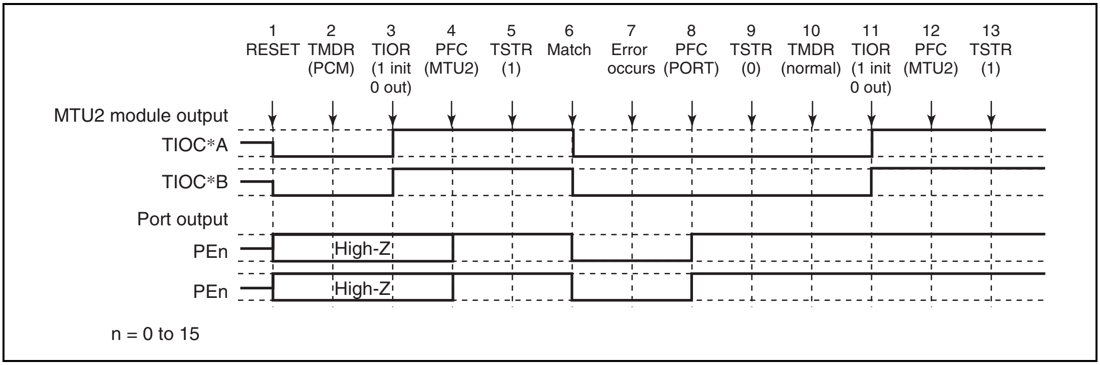


Figure 11.155 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set MTU2 output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

17.3.6 Serial Control Register (SCSCR)

SCSCR operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE[1:0]	
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/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), when the quantity of data in the transmit FIFO register becomes less than the specified number of transmission triggers, and when the TDFE flag in the serial status register (SCFSR) is set to 1. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled* Note: * The TXI interrupt request can be cleared by writing a greater quantity of transmit data than the specified transmission trigger number to SCFTDR and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.

17.4 Operation

17.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a clocked synchronous mode in which communication is synchronized with clock pulses.

The SCIF has a 16-stage FIFO buffer for both transmission and receptions, reducing the overhead of the CPU, and enabling continuous high-speed communication.

The transmission format is selected in the serial mode register (SCSMR), as shown in table 17.14. The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 17.15.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

(2) Clocked Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of the on-chip baud rate generator, and outputs this clock to external devices as the synchronous clock.
  - When an external clock is selected, the SCIF operates on the input synchronous clock not using the on-chip baud rate generator.

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I^2C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. When seven bits after the start condition is issued in slave receive mode match the slave address set to SAR and the 8th bit is set to 1, TRS is automatically set to 1. If an overrun error occurs in master receive mode with the clocked synchronous serial format, MST is cleared and the mode changes to slave receive mode. Operating modes are described below according to MST and TRS combination. When clocked synchronous serial format is selected and MST = 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3 to 0	CKS[3:0]	0000	R/W	Transfer Clock Select These bits should be set according to the necessary transfer rate (table 19.3) in master mode.

- Port A Control Register L3 (PACRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA11MD[2:0]			-	PA10MD[2:0]			-	PA9MD[2:0]			-	PA8MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PA11MD[2:0]	000*	R/W	PA11 Mode Select the function of the PA11/$\overline{\text{CS1}}$/IRQ1/TIC5V/CRx0/RXD0/TX_EN pin. 000: PA11 I/O (port) 001: $\overline{\text{CS1}}$ output (BSC) 010: Setting prohibited 011: IRQ1 input (INTC) 100: TIC5V input (MTU2) 101: CRx0 input (RCAN-ET) 110: RXD0 input (SCI) 111: TX_EN input (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PA10MD[2:0]	000*	R/W	PA10 Mode Select the function of the PA10/$\overline{\text{CS2}}$/IRQ2/TIC5W/CTx0/TXD0/MII_TXD0 pin. 000: PA10 I/O (port) 001: $\overline{\text{CS2}}$ output (BSC) 010: Setting prohibited 011: IRQ2 input (INTC) 100: TIC5W input (MTU2) 101: CTx0 output (RCAN-ET) 110: TXD0 output (SCI) 111: MII_TXD0 output (Ether)

25.3.18 Multicast Address Frame Receive Counter Register (MAFCR)

MAFCR is a 32-bit counter that indicates the number of received frames that specify a multicast address. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MAFC[31:16]															
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
	MAFC[15:0]															
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	Bit Name	Initial Value	R/W	Description
31 to 0	MAFC[31:0]	All 0	R/W	Multicast Address Frame Count These bits indicate the number of multicast address frames received.

25.3.23 Random Number Generation Counter Upper Limit Register (RDMLR)

RDMLR is used to set the upper limit for the counter used in the random number generation block.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	RMD[19:16]			
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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMD[15:0]															
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	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 0	RMD[19:0]	All 0	R/W	Upper Limit for Counter Used in Random Number Generation Block H'00000: Used in normal operation H'00001to H'FFFFE: Upper limit for the counter

Note: The setting of this register affects the operation of the random number generation block in the feLic. Pay attention when setting a value other than H'00000.

- Programming/erasing unit

  The data MAT is programmed in 8-byte or 128-byte units and erased in block units (8 Kbytes) in user mode, user program mode, and user boot mode. In boot mode, the data MAT is programmed in 256-Kbyte units and erased in block units (8 Kbytes). The product information MAT is read-only memory and cannot be programmed or erased.

  Figure 28.3 shows the block configuration of the data MAT of this LSI. The data MAT is divided into four 8-Kbyte blocks (DB00 to DB03).

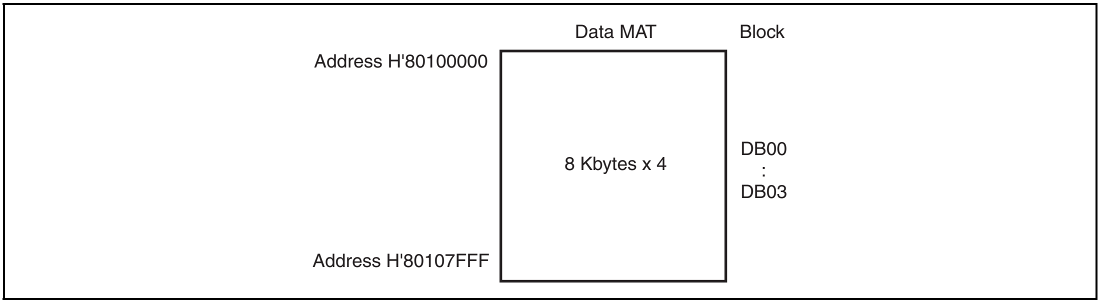


Figure 28.3 Block Configuration of Data MAT

- Blank check function

  If data is read from erased FLD by the CPU, undefined values are read. Using blank check command of the FCU allows checking of whether the FLD is erased (in a blank state). Either an 8 Kbytes (1 erasure block) or 8 bytes of area can be checked by a single execution of the blank check command.

  Blank checking proceeds for areas where erasure has been completed normally to confirm that the data have actually been erased. When erasure or programming in progress is stopped (e.g. by input of the reset signal or shutting down the power), blank checking cannot be used to check whether the data have actually been erased or written.

- Four types of on-board programming modes
  - Boot mode

    The data MAT can be programmed using the SCI. The bit rate for SCI communications between the host and the LSI can be automatically adjusted.
  - User mode/user program mode

    The data MAT can be programmed with a desired interface. The user mode includes the MCU extended mode and MCU single-chip mode (modes 2 and 3) in which the on-chip ROM is enabled.

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>27.10.10 Items Prohibited during Programming and Erasure</td><td>1575</td><td>Added
• Cutting off the power supply
• Transitions to software standby mode
• Read access to the flash memory by the CPU, DMAC or DTC
• Writing a new value to the FRQCR register
• Setting the PCKAR register for a different frequency from that of $P\phi$.</td></tr>
<tr><td>27.10.11 Abnormal Ending of Programming or Erasure</td><td>1575</td><td>Subsection added</td></tr>
<tr><td>28.1 Features
• Blank check function</td><td>1580</td><td>Added
Blank checking proceeds for areas where erasure has been completed normally to confirm that the data have actually been erased. When erasure or programming in progress is stopped (e.g. by input of the reset signal or shutting down the power), blank checking cannot be used to check whether the data have actually been erased or written.</td></tr>
<tr><td>28.8.9 Items Prohibited during Programming and Erasure</td><td>1619</td><td>Added
• Cutting off the power supply
• Transitions to software standby mode
• Read access to the flash memory by the CPU, DMAC or DTC
• Writing a new value to the FRQCR register
• Setting the PCKAR register for a different frequency from that of $P\phi$.</td></tr>
<tr><td>28.8.10 Abnormal Ending of Programming or Erasure
28.8.11 Handling when Erasure or Programming is Stopped</td><td>1619</td><td>Subsection added</td></tr>
<tr><td>Table 30.4 Register States in Software Standby Mode</td><td>1641</td><td>Amended
<table>
<tr><th>Module Name</th><th>Initialized Registers</th><th>Registers Whose Content is Retained</th></tr>
<tr><td>Compare match timer (CMT)</td><td>—</td><td>All registers</td></tr>
</table>
</td></tr>
</table>