Toshiba Semiconductor and Storage - TMPM333FYFG(C) 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	Active
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	40MHz
Connectivity	I²C, SIO, UART/USART
Peripherals	POR, WDT
Number of I/O	78
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmpm333fyfg-c

Table 1-1 Pin Names and Functions Sorted by Pin (3/6)

Type	Pin No.	Pin Name	Input/ Output	Function
Function	36	PH3 TB1IN1	I/O I	I/O port Inputting the timer B capture trigger
Function	37	PB4	I/O	I/O port
Function	38	PI0 TB0OUT	I/O O	I/O port Timer B output
Function	39	PJ6 INT6	I/O I	I/O port External interrupt pin
Function	40	PI1 TB1OUT	I/O O	I/O port Timer B output
Function	41	PB5	I/O	I/O port
Function	42	PI2 TB2OUT	I/O O	I/O port Timer B output
Function	43	PB6	I/O	I/O port
Function	44	PF4 SDA1/SO1	I/O I/O	I/O port -in the I2C mode: data pin -in the SIO mode: data pin
Function	45	PF5 SCL1/SI1	I/O I/O	I/O port -in the I2C mode: clock pin -in the SIO mode: data pin
Function	46	PF6 SCK1	I/O I/O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
Function	47	PB7	I/O	I/O port
Function	48	PI3 TB3OUT	I/O O	I/O port Timer B output
Function	49	PJ1 INT1	I/O I	I/O port External interrupt pin
Function	50	PK0	I/O	I/O port (note) Nch open drain port.
Function	51	PK1 SCOUT $\overline{\text{ALARM}}$	I/O O O	I/O port System clock output Alarm output
Function	52	PI4 TB4OUT	I/O O	I/O port Timer B output
Function	53	PI5 TB5OUT	I/O O	I/O port Timer B output
Function/ Debug	54	PB0 TDO/SWV	I/O O	I/O port Debug pin
Function/ Debug	55	PA0 TMS/SWDIO	I/O I/O	I/O port Debug pin

4.1.2 Memory Map of TMPM333FYFG

Figure 4-2 shows the memory map of the TMPM333FYFG.

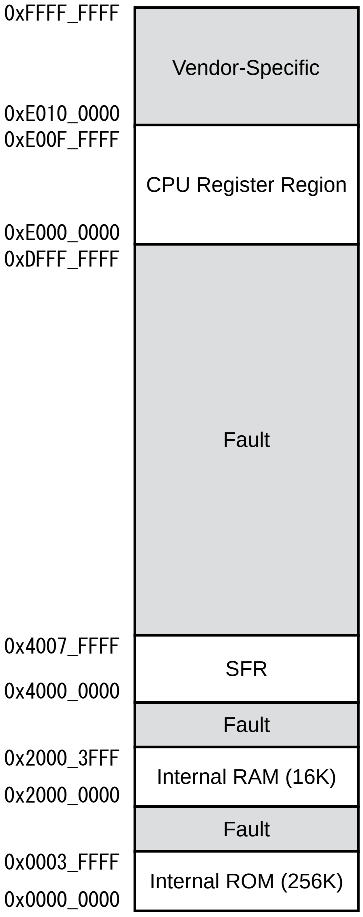


Figure 4-2 Memory Map (TMPM333FYFG)

Note: In addition to 256KB flash area, the TMPM333FYFG provides 128-word data/ password area (1 page) for Show Product Information command in the address range 0x0007_FE00 - 0x0007_FFFF. See the Chapter "Flash Memory Operation" for details on the single boot mode.

Do not Access to the range from 0x0004_0000 through the password area.

6.6.6 Releasing the Low Power Consumption Mode

The low power consumption mode can be released by an interrupt request, Non-Maskable Interrupt (NMI) or reset. The release source that can be used is determined by the low power consumption mode selected.

Details are shown in Table 6-8.

Table 6-8 Release Source in Each Mode

Low power consumption mode			IDLE	SLEEP	STOP
Release source	Interrupt	INT0 to 7 (Note1)	o	o	o
		INTRTC	o	o	×
		INTTB0 to 9	o	×	×
		INTCAP00 to 60, 01 to 61	o	×	×
		INTRX0 to 2, INTTX0 to 2	o	×	×
		INTSBI0 to 2	o	×	×
		INTAD/INTADHP/INTADM0, 1	o	×	×
	NMI (INTWDT)		o	×	×
	NMI ($\overline{\text{NMI}}$ pin)		o	o	o
	RESET ($\overline{\text{RESET}}$ pin)		o	o	o

o : Starts the interrupt handling after the mode is released. (The reset initializes the LSI)

× : Unavailable

Note 1: **To release the low power consumption mode by using the level mode interrupt, keep the level until the interrupt handling is started. Changing the level before then will prevent the interrupt handling from starting properly.**

Note 2: **For shifting to the low power consumption mode, set the CPU to prohibit all the interrupts other than the release source. If not, releasing may be executed by an unspecified interrupt.**

- Release by interrupt request

  To release the low power consumption mode by an interrupt, the CPU must be set in advance to detect the interrupt. In addition to the setting in the CPU, the clock generator must be set to detect the interrupt to be used to release the SLEEP and STOP modes.

- Release by Non-Maskable Interrupt (NMI)

  There are two kinds of NMI sources: WDT interrupt (INTWDT) and NMI pin. INTWDT can only be used in the IDLE mode. The NMI pin can be used to release all the lower power consumption modes.

- Release by reset

  Any low power consumption mode can be released by reset from the $\overline{\text{RESET}}$ pin. After that, the mode switches to the NORMAL mode and all the registers are initialized as is the case with normal reset.

  Note that returning to the STOP mode by reset does not induce the automatic warm-up. Keep the reset signal valid until the oscillator operation becomes stable.

Refer to "Interrupts" for details.

Processing	Details	See
ISR execution	Program for the ISR. Clear the interrupt source if needed.	"7.5.2.6 Interrupt Service Routine (ISR)"
⇩		
Return to preceding program	Configure to return to the preceding program of the ISR.	

7.5.2.2 Preparation

When preparing for an interrupt, you need to pay attention to the order of configuration to avoid any unexpected interrupt on the way.

Initiating an interrupt or changing its configuration must be implemented in the following order basically. Disable the interrupt by the CPU. Configure from the farthest route from the CPU. Then enable the interrupt by the CPU.

To configure the clock generator, you must follow the order indicated here not to cause any unexpected interrupt. First, configure the precondition. Secondly, clear the data related to the interrupt in the clock generator and then enable the interrupt.

The following sections are listed in the order of interrupt handling and describe how to configure them.

1. Disabling interrupt by CPU
2. CPU registers setting
3. Preconfiguration (1) (Interrupt from external pin)
4. Preconfiguration (2) (Interrupt from peripheral function)
5. Preconfiguration (3) (Interrupt Set-Pending Register)
6. Configuring the clock generator
7. Enabling interrupt by CPU

(1) Disabling interrupt by CPU

To make the CPU for not accepting any interrupt, write "1" to the corresponding bit of the PRIMASK Register. All interrupts and exceptions other than non-maskable interrupts and hard faults can be masked.

Use "MSR" instruction to set this register.

Interrupt mask register		
PRIMASK	←	"1" (interrupt disabled)

Note 1: PRIMASK register cannot be modified by the user access level.

Note 2: **If a fault causes when "1" is set to the PRIMASK register, it is treated as a hard fault.**

(2) CPU registers setting

You can assign a priority level by writing to <PRI_n> field in an Interrupt Priority Register of the NVIC register.

7.6.2.17 System Handler Control and State Register

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SVCALL PENDED	BUSFAULT PENDED	MEMFAULT PENDED	USGFAULT PENDED	SYSTICKACT	PENDSVACT	-	MONITOR ACT
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SVCALLACT	-	-	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-19	−	R	Read as 0.
18	USGFAULT ENA	R/W	Usage Fault 0: Disabled 1: Enable
17	BUSFAUL TENA	R/W	Bus Fault 0: Disabled 1: Enable
16	MEMFAULT ENA	R/W	Memory Management 0: Disabled 1: Enable
15	SVCALL PENDED	R/W	SVCall 0: Not pended 1: Pended
14	BUSFAULT PENDED	R/W	Bus Fault 0: Not pended 1: Pended
13	MEMFAULT PENDED	R/W	Memory Management 0: Not pended 1: Pended
12	USGFAULT PENDED	R/W	Usage Fault 0: Not pended 1: Pended
11	SYSTICKACT	R/W	SysTick 0: Inactive 1: Active
10	PENDSVACT	R/W	PendSV 0: Inactive 1: Active
9	−	R	Read as 0.
8	MONITORACT	R/W	Debug Monitor 0: Inactive 1: Active
7	SVCALLACT	R/W	SVCall 0: Inactive 1: Active
6-4	−	R	Read as 0.

8.2.5.3 PEDATA (Port E data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6	PE5	PE4	PE3	PE2	PE1	PE0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PE6-PE0	R/W	Port E data register

8.2.5.4 PECR (Port E output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PE6C-PE0C	R/W	Output 0: Disable 1: Enable

8.2.6.5 PFFR1(Port F function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7F1	PF6F1	PF5F1	PF4F1	PF3F1	PF2F1	PF1F1	PF0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PF7F1	R/W	0: PORT 1: INT5
6	PF6F1	R/W	0: PORT 1: SCK1
5	PF5F1	R/W	0: PORT 1: SI1/SCL1
4	PF4F1	R/W	0: PORT 1: SO1/SDA1
3	PF3F1	R/W	0: PORT 1: Reserved
2	PF2F1	R/W	0: PORT 1: SCLK2
1	PF1F1	R/W	0: PORT 1: RXD2
0	PF0F1	R/W	0: PORT 1: TXD2

8.3.19 Type T18

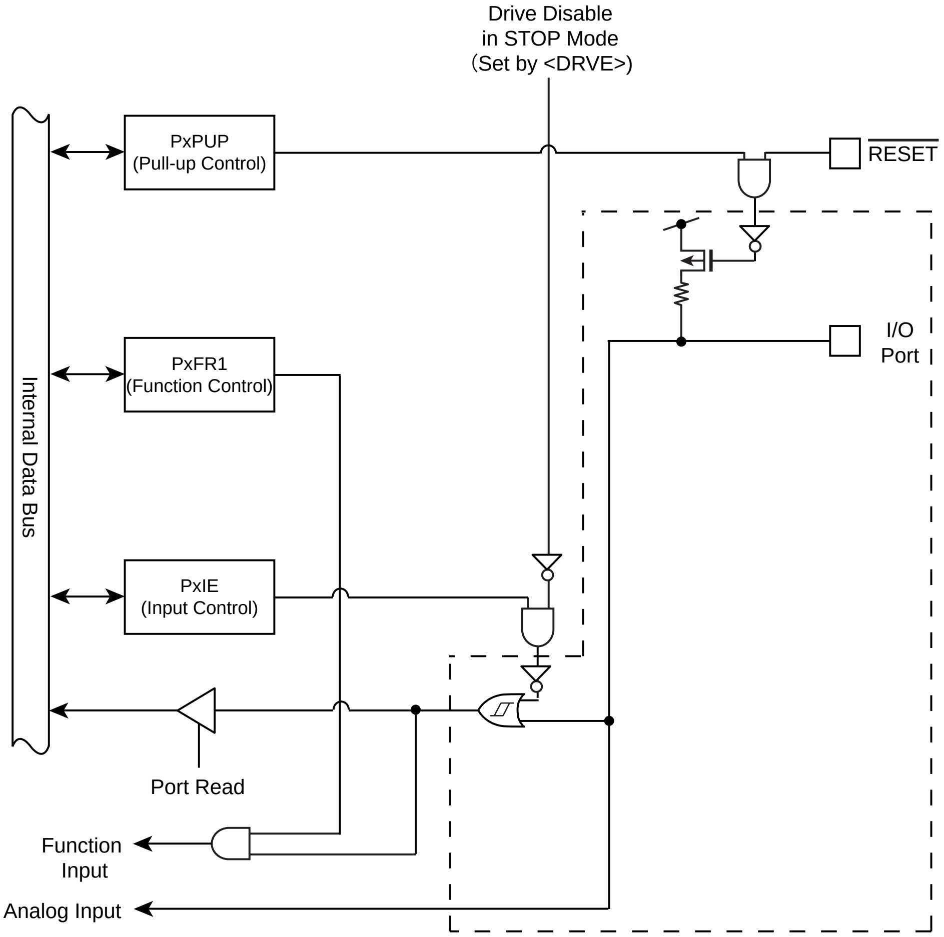


Figure 8-18 Port TypeT18

8.4.7 Port G Setting

Table 8-12 Port Setting List (Port G)

Pin	Port Type	Function	After re-set	PGCR	PGFR1	PGOD	PGPUP	PGIE
PG0	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SO0(Output)		1	1	x	x	0
		SDA0(Input/Output)		1	1	1	x	1
PG1	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SI0(Input)		0	1	x	x	1
		SCL0(Input/Output)		1	1	1	x	1
PG2	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SCK0(Input)		0	1	x	x	1
		SCK0(Output)		1	1	x	x	0
PG3	T8	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		INT4(Input)		0	1	x	x	1
PG4	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SO2(Output)		1	1	x	x	0
		SDA2(Input/Output)		1	1	1	x	1
PG5	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SI2(Input)		0	1	x	x	1
		SCL2(Input/Output)		1	1	1	x	1
PG6	T13	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		SCK2(Input)		0	1	x	x	1
		SCK2(Output)		1	1	x	x	0
PG7	T10	Input Port		0	0	x	x	1
		Output Port		1	0	x	x	0
		TB8OUT(Output)		1	1	x	x	0

9.6 Description of Operations for Each Mode

9.6.1 16-bit Interval Timer Mode

In the case of generating constant period interrupt, set the interval time to the Timer register (TBxRG1) to generate the INTTBx interrupt.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits INTTBx interrupt by setting corresponding bit to "1".
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger
TBxMOD	←	X	0	1	0	0	1	*	*	Changes to prescaler output clock as input clock. Specifies Capture function to disable.
							(** = 01, 10, 11)			
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don’t care
−; No change

9.6.2 16-bit Event Counter Mode

It is possible to make it the event counter by using an input clock as an external clock (TBxIN0 pin input).

The up-counter counts up on the rising edge of TBxIN0 pin input. It is possible to read the count value by capturing value using software and reading the captured value.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
PxIE[m]	←								1	Allocates corresponding port toTBxIN0.
PxFR1[m]	←								1	
TBxFFCR	←	X	X	0	0	0	0	1	1	Disables to TBxFF0 reverse trigger
TBxMOD	←	X	0	1	0	0	0	0	0	Changes toTBxIN0 as an input clock
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.
TBxMOD	←	X	0	0	0	0	0	0	0	Software capture is done.

Note 1: m: corresponding bit of port
Note 2: X; Don’t care
−; No change

Table 10-4 Clock Resolution to the Baud Rate Generator fc = 40 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.05 μs)	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)
		001 (fperiph/2)	fc/2^2 (0.1 μs)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)
		010 (fperiph/4)	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		011 (fperiph/8)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)
		001 (fperiph/2)	fc/2^2 (0.1 μs)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)
		010 (fperiph/4)	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		011 (fperiph/8)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	101 (fc/4)	000 (fperiph/1)	–	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)
		001 (fperiph/2)	–	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)
		010 (fperiph/4)	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		011 (fperiph/8)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)
		001 (fperiph/2)	–	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)
		010 (fperiph/4)	–	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		011 (fperiph/8)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)

Note 1: **The prescaler output clock φTn must be selected so that the relationship "φTn ≤ fsys / 2" is satisfied (so that φTn is slower than fsys).**

Note 2: **Do not change the clock gear while SIO is operating.**

Note 3: **The dashes in the above table indicate that the setting is prohibited.**

10.14.2 TX interrupts

Figure 10-11 shows the data flow of transmit operation and the route of read.

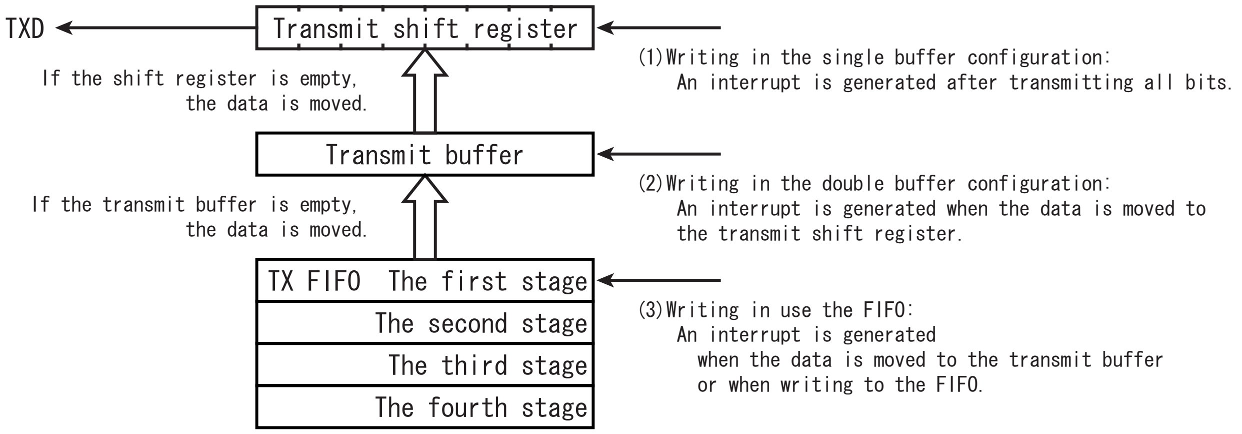

TXD
Transmit shift register
If the shift register is empty,
the data is moved.
Transmit buffer
If the transmit buffer is empty,
the data is moved.
TX FIFO The first stage
The second stage
The third stage
The fourth stage
(1)Writing in the single buffer configuration:
An interrupt is generated after transmitting all bits.
(2)Writing in the double buffer configuration:
An interrupt is generated when the data is moved to
the transmit shift register.
(3)Writing in use the FIFO:
An interrupt is generated
when the data is moved to the transmit buffer
or when writing to the FIFO.

Figure 10-11 Transmit Buffer/FIFO Configuration Diagram

10.14.2.1 Single Buffer / Double Buffer

TX interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Buffer Configurations	UART modes	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register.	

Note: If double buffer is enabled, a interrupt is also generated when the data is moved from the buffer to the shift register by writing to the buffer.

10.14.2.2 FIFO

In use of FIFO, transmit interrupt is generated on the condition that the fllowing either operation and SCxTFC<TFIS> setting are established.

- Transmittion completion of all bits of one frame.
- Writing FIFO

Interrupt conditions are decided by the SCxTFC<TFIS> settings as described in Table 10-13.

Table 10-13 Transmit Interrupt conditions in use of FIFO

SCxTFC<TFIS>	Interrupt conditions
"0"	"The fill level of FIFO" is equal to "the fill level of FIFO interruption generation."
"1"	"The fill level of FIFO" is smaller than or equal to "the fill level of FIFO intrruption generation."

11.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBIxI2CAR<ALS>="0"), SBIxSR<AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBIxI2CAR.

When <ALS> is "1", <AAS> is set to "1" when the first data word has been received. <AAS> is cleared to "0" when data is written to or read from SBIxDBR.

11.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBIxSR<AD0> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros.

<AD0> is cleared to "0" when the start or stop condition is detected on the bus.

11.5.13 Last Received Bit Monitor

SBIxSR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBIxSR<LRB> immediately after generation of the INTSBIx interrupt request causes ACK signal to be read.

11.5.14 Data Buffer Register (SBIxDBR)

Reading or writing SBIxDBR initiates reading received data or writing transmitted data.

When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

11.5.15 Baud Rate Register (SBIxBR0)

The SBIxBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode.

This register must be programmed before executing an instruction to switch to the standby mode.

11.5.16 Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing "10" followed by "01" to SBIxCR2<SWRST[1:0]> generates a reset signal that initializes the serial bus interface circuit. After a reset, all control registers and status flags are initialized to their reset values. When the serial bus interface is initialized, <SWRST> is automatically cleared to "0".

Note: A software reset causes the SBI operating mode to switch from the I2C mode to the port mode.

12.2 Configuration

Figure 12-1 shows the block diagram of this AD converter.

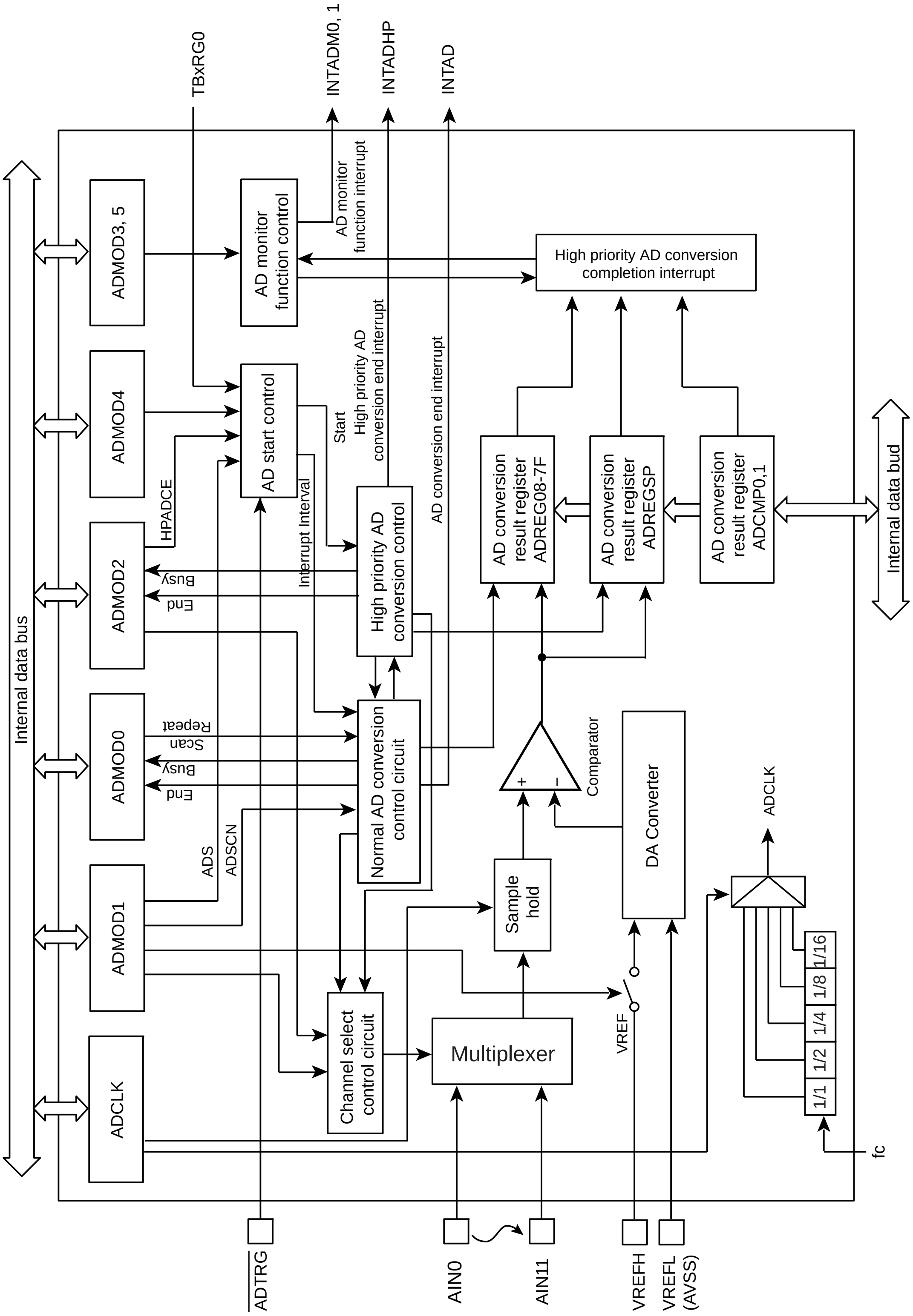

TBxRG0
INTADM0, 1
INTADHP
INTAD
Internal data bus
ADMOD3, 5
ADMOD4
ADMOD2
ADMOD0
ADMOD1
ADCLK
AD monitor function control
AD monitor function interrupt
High priority AD conversion completion interrupt
AD start control
Start
High priority AD conversion end interrupt
AD conversion end interrupt
HPADCE
Interrupt Interval
Busy
End
High priority AD conversion control
Repeat
Scan
Busy
End
Normal AD conversion control circuit
ADS
ADSCN
Channel select control circuit
Multiplexer
Sample hold
Comparator
+
-
DA Converter
VREF
AD conversion result register ADREG08-7F
AD conversion result register ADREGSP
AD conversion result register ADCMP0,1
Internal data bud
ADCLK
1/1
1/2
1/4
1/8
1/16
fc
ADTRG
AIN0
AIN11
VREFH
VREFL (AVSS)

Figure 12-1 AD Converter Block Diagram

12.3.8 ADMOD3 (Mode Control Register 3)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	ADOBIC0	ADREGS0				ADOBSV0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	−	R/W	Write "0".
6	−	R	Read as 0.
5	ADOBIC0	R/W	Set the AD monitor function interrupt 0 0: If the value of the conversion result is smaller than the comparison register 0, an interrupt is generated. 1: If the value of the conversion result is bigger than the comparison register 0, an interrupt is generated.
4-1	ADREGS0[3:0]	R/W	Select a target conversion result register when using the AD monitor function 0 (See the below table).
0	ADOBSV0	R/W	AD monitor function 0 0: Disable 1: Enable

<ADREGS0[3:0]>	Conversion result register to be compared	<ADREGS0[3:0]>	Conversion result register to be compared
0000	ADREG08	0100	ADREG4C
0001	ADREG19	0101	ADREG5D
0010	ADREG2A	0110	ADREG6E
0011	ADREG3B	0111	ADREG7F
-	-	1xxx	ADREGSP

13.3 Operations

13.3.1 Basic Operation

The Watchdog timer is consists of the binary counters that work using the system clock (fsys) as an input. Detecting time can be selected between 2^{15}, 2^{17}, 2^{19} , 2^{21}, 2^{23} and 2^{25} by the WDMOD<WDTP[2:0]>. The detecting time as specified is elapsed, the watchdog timer interrupt (INTWDT) generates, and the watchdog timer out pin ($\overline{\text{WDTOUT}}$) output "Low".

To detect malfunctions (runaways) of the CPU caused by noise or other disturbances, the binary counter of the watchdog timer should be cleared by software instruction before INTWDT interrupt generates. If the binary counter is not cleared, the non-maskable interrupt generates by INTWDT. Thus CPU detects malfunction (runway), malfunction countermeasure program is performed to return to the normal operation.

Additionally, it is possible to resolve the problem of a malfunction (runaway) of the CPU by connecting the watchdog timer out pin to reset pins of peripheral devices.

Note: This product does not include a watchdog timer out pin ($\overline{\text{WDTOUT}}$).

13.3.2 Operation Mode and Status

The watchdog timer begins operation immediately after a reset is cleared.

If not using the watchdog timer, it should be disabled.

The watchdog timer cannot be used in the STOP mode, SLEEP mode and SLOW mode where high-speed frequency clock is stopped. Before transition to these modes, the watchdog timer should be disabled.

In IDLE mode, its operation depends on the WDMOD <I2WDT> setting.

Also, the binary counter is automatically stopped during debug mode.

14.5.2 1Hz cycle "Low" pulse1 Hz

The RTC outputs a "Low" pulse cycle of low-speed 1Hz clock to the $\overline{\text{ALARM}}$ pin by setting RTCPAGER<INTENA>="1" after setting RTCPAGER<ENAALM>= "0", RTCRESTR<DIS1HZ>= "0" and <DIS16HZ>= "1". It generates an INTRTC interrupt simultaneously.

14.5.3 16Hz cycle "Low" pulse16 Hz

The RTC outputs a "Low" pulse cycle of low-speed 16Hz clock to the $\overline{\text{ALARM}}$ pin by setting RTCPAGER<INTENA>="1" after setting RTCPAGER<ENAALM>= "0", RTCRESTR<DIS1HZ>= "1" and <DIS16HZ>= "0". It generates an INTRTC interrupt simultaneously.

15.2.2.2 (1-B) Method 2: Transferring a Programming Routine from an External Host

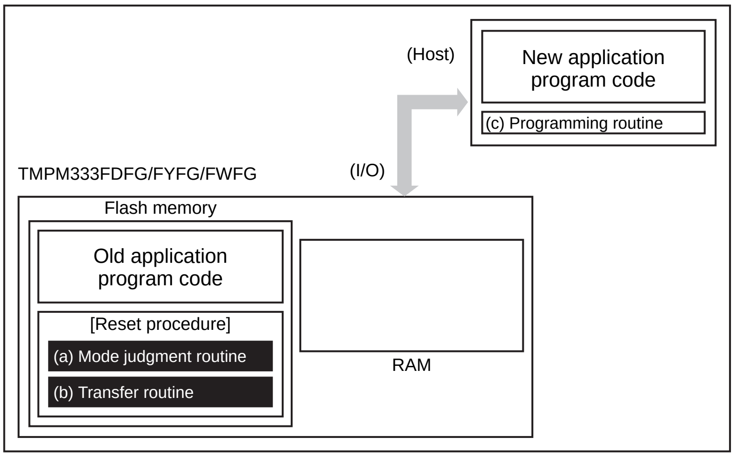
(1) Step-1

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM333FDFG/FYFG/FWFG on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine:	Code to determine whether or not to switch to User Boot mode
(b) Transfer routine:	Code to download new program code from a host controller

Also, prepare a programming routine shown below on the host controller:

(c) Programming routine:	Code to download new program code from an external host controller and re-program the flash memory



16.4 Writing and erasing

16.4.1 Protection bits

Writing and erasing protection bits are available with a single chip mode, single boot mode and writer mode.

Writing to the protection bits is done on block-by-block basis.

Erasing of the protection bits is done by two groups of the blocks: block 0 through 3 and block 4 through 5. When the settings for all the blocks are "1", erasing must be done after setting the FCSECBIT <SECBIT> bit to "0". Setting "1" at that situation erases all the protection bits. To write and erase the protection bits, command sequence is used.

See the capter "Flash" for details

16.4.2 Security bit

The FCSECBIT <SECBIT> bit that activates security function is set to "1" at a power-on reset right after power-on.

The bit is rewritten by the following procedure.

1. Write the code 0xa74a9d23 to FCSECBIT register.
2. Write data within 16 clocks from the above.1.

Note: The above procedure is enabled only when using 32-bit data transfer command.