Toshiba Semiconductor and Storage - TMPM330FDFG(C) Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	40MHz
Connectivity	I²C, SIO, UART/USART
Peripherals	POR, WDT
Number of I/O	79
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K 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/tmpm330fdfg-c

10. Serial Channel (SIO/UART)

6. Clock/Mode control

6.1 Features

The clock/mode control block enables to select clock gear, prescaler clock and warm-up of the PLL clock multiplication circuit and oscillator.

There is also the low power consumption mode which can reduce power consumption by mode transitions.

This chapter describes how to control clock operating modes and mode transitions.

The clock/mode control block has the following functions:

- Controls the system clock
- Controls the prescaler clock
- Controls the PLL multiplication circuit
- Controls the warm-up timer

In addition to NORMAL mode, the TMPM330FDFG/FYFG/FWFG can operate in three types of low power mode to reduce power consumption according to its usage conditions.

Each interrupt source is provided with eight bits for assigning a priority level from 0 to 255, but the number of bits actually used varies with each product.Priority level 0 is the highest priority level.If multiple sources have the same priority, the smallest-numbered interrupt source has the highest priority.

You can assign grouping priority by using the PRIGROUP field in the Application Interrupt and Reset Control Register.

NVIC register		
<PRI_n>	←	"priority"
<PRIGROUP>	←	"group priority"(This is configurable if required.)

Note: **"n" indicates the corresponding exceptions/interrupts.**
This product uses three bits for assigning a priority level.

(3) Preconfiguration (1) (Interrupt from external pin)

Set "1" to the port function register of the corresponding pin. Setting PxFRn[m] allows the pin to be used as the function pin. Setting PxIE[m] allows the pin to be used as the input port.

Port register		
PxFRn<PxmFn>	←	"1"
PxIE<PxmIE>	←	"1"

Note: **x: port number / m: corresponding bit / n: function register number**
In modes other than STOP mode, setting PxIE to enable input enables the corresponding interrupt input regardless of the PxFR setting. Be careful not to enable interrupts that are not used. Also, be aware of the description of "7.5.1.4 Precautions when using external interrupt pins".

(4) Preconfiguration (2) (Interrupt from peripheral function)

The setting varies depending on the peripheral function to be used. See the chapter of each peripheral function for details.

(5) Preconfiguration (3) (Interrupt Set-Pending Register)

To generate an interrupt by using the Interrupt Set-Pending Register, set "1" to the corresponding bit of this register.

NVIC register		
Interrupt Set-Pending [m]	←	"1"

Note: **m: corresponding bit**

(6) Configuring the clock generator

For an interrupt source to be used for exiting a standby mode, you need to set the active level and enable interrupts in the CGIMCG register of the clock generator. The CGIMCG register is capable of configuring each source.

8.2.1.3 PADATA (Port A 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	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PA7-PA0	R/W	Port A data register.

8.2.1.4 PACR (Port A 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	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PA7C-PA0C	R/W	Output 0: disable 1: enable

8.2.7.6 PGOD (Port G open drain 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	PG7OD	PG6OD	PG5OD	PG4OD	PG3OD	PG2OD	PG1OD	PG0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7OD- PG0OD	R/W	0: CMOS 1: Open-drain

8.2.7.7 PGPUP (Port G pull-up 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	PG7UP	PG6UP	PG5UP	PG4UP	PG3UP	PG2UP	PG1UP	PG0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7UP- PG0UP	R/W	Pull-up 0: Disable 1: Enable

8.3.9 Type T8

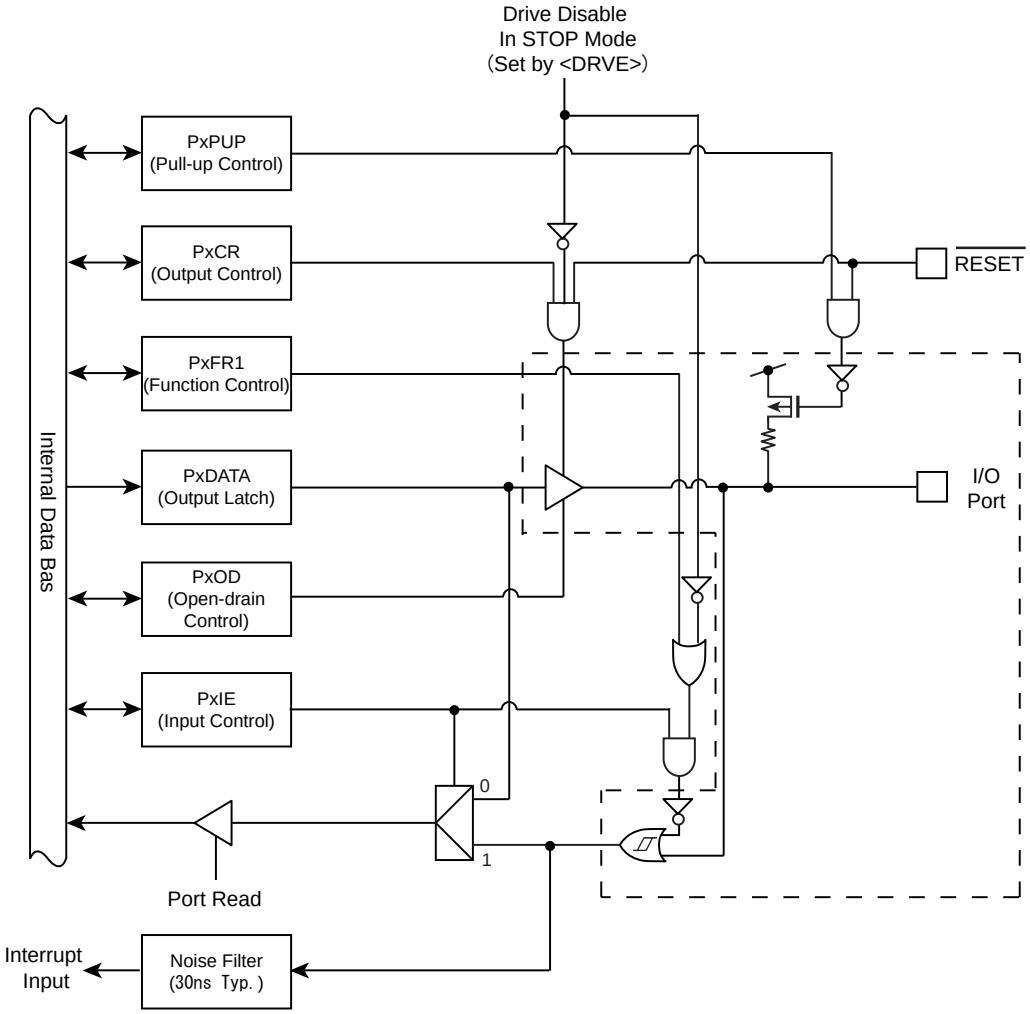


Figure 8-8 Port Type T8

Table 9-3 Prescaler Output Clock Resolutions(fc = 32MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.0625 μs)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)
		001 (fperiph/2)	fc/2^2 (0.125 μs)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
	100 (fc/2)	000 (fperiph/1)	−	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)
		001 (fperiph/2)	fc/2^2 (0.125 μs)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
	101 (fc/4)	000 (fperiph/1)	−	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)
		001 (fperiph/2)	−	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)
		010 (fperiph/4)	fc/2^3 (0.25 μs)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)
	110 (fc/8)	000 (fperiph/1)	−	−	fc/2^5 (1.0 μs)
		001 (fperiph/2)	−	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)
		010 (fperiph/4)	−	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)
		011 (fperiph/8)	fc/2^4 (0.5 μs)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)
		100 (fperiph/16)	fc/2^5 (1.0 μs)	fc/2^7 (4.0 μs)	fc/2^9 (16.0 μs)
		101 (fperiph/32)	fc/2^6 (2.0 μs)	fc/2^8 (8.0 μs)	fc/2^{10} (32.0 μs)

Note 1: The prescaler output clock φTn must be selected so that φTn < fsys is satisfied (so that φTn is slower than fsys).

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

Note 3: "−" denotes a setting prohibited.

10.4.5 SCxMOD0 (Mode Control Register 0)

	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	TB8	CTSE	RXE	WU	SM		SC	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	TB8	R/W	Transmit data bit 8 (For UART) Writes the 9th bit of transmit data in the 9 bits UART mode.
6	CTSE	R/W	Handshake function control (For UART) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{CTS}$ pin.
5	RXE	R/W	Receive control (Note) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For UART) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. In it is Enabled, Interrupt only when RB9 = "1" at 9-bit UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For UART) 00: Timer TB9OUT 01: Baud rate generator 10: Internal clock fsys 11: External clock (SCLK input) (As for the I/O interface mode, the serial transfer clock can be set in the control register (SCxCR).

Note 1: **With <RXE> set to "0", set each mode register (SCxMOD0, SCxMOD1 and SCxMOD2). Then set <RXE> to "1".**

Note 2: Do not stop the receive operation (by setting SCxMOD0<RXE> = "0") when data is being received.

10.6.2 Parity Control

The parity bit can be added only in the 7- or 8-bit UART mode.

Setting "1" to SCxCR<PE> enables the parity.

The <EVEN> bit of SCxCR selects either even or odd parity.

10.6.2.1 Transmission

Upon data transmission, the parity control circuit automatically generates the parity with the data in the transmit buffer.

After data transmission is complete, the parity bit will be stored in SCxBUF<TB7> in the 7-bit UART mode and SCxMOD<TB8> in the 8-bit UART mode.

The <PE> and <EVEN> settings must be completed before data is written to the transmit buffer.

10.6.2.2 Receiving Data

If the received data is moved from the receive shift register to the receive buffer, a parity is generated.

In the 7-bit UART mode, the generated parity is compared with the parity stored in SCxBUF<RB7>, while in the 8-bit UART mode, it is compared with the one in SCxCR<RB8>.

If there is any difference, a parity error occurs and the <PERR> of the SCxCR register is set to "1".

In use of the FIFO, <RERR> indicates that a parity error was generated in one of the recieved data.

10.6.3 STOP Bit Length

The length of the STOP bit in the UART transmission mode can be selected from one bit or two bits by setting the SCxMOD2<SBLEN>. The length of the STOP bit data is determined as one-bit when it is received regardless of the setting of this bit.

(2) SCLK Input Mode

- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  If the SCLK is input in the condition where data is written in the transmit buffer, 8-bit data is outputted from the TXD pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing point "A" as shown in Figure 10-13.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer before the SCLK input becomes active or when data transmission from the transmit shift register is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  If the SCLK input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (0xFF) is sent.

11.4.3 SBIxCR2(Control register 2)

This register serves as SBIxSR register by reading it.

	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	MST	TRX	BB	PIN	SBIM		SWRST	
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBIx interrupt request 0: − 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note) 00: Port mode (Disables a serial bus interface output) 01: SIO mode 10: I2C bus mode 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset.

Note: **Make sure that modes are not changed during a communication session.Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "High" level before switching the operating mode from the port mode to the I2C bus or clock-synchronous 8-bit SIO mode.**

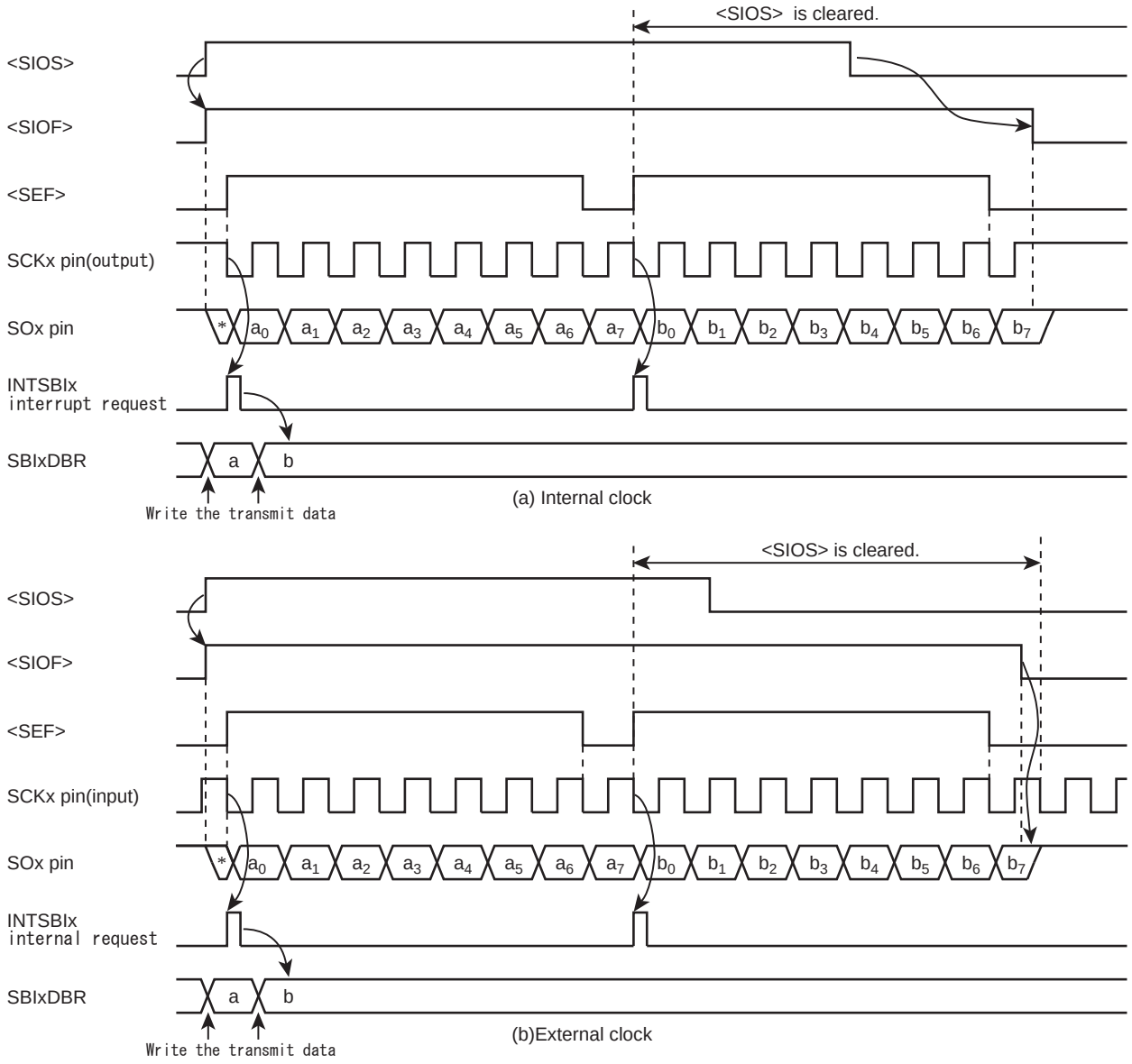


Figure 11-18 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

```
                                  7  6  5  4  3  2  1  0
if SBIxSR<SIOF> ≠ 0                                        Recognizes the completion of the transmission.
Then
if SCK ≠ 1                                                 Recognizes "1" is set to the SCK pin by monitoring the port.
Then
SBIxCR1          ←  0  0  0  0  0  1  1  1                 Completes the transmission by setting <SIOS> = 0.
```

12.2 Block Diagram

Figure 12-1 shows the Block Diagram of CEC

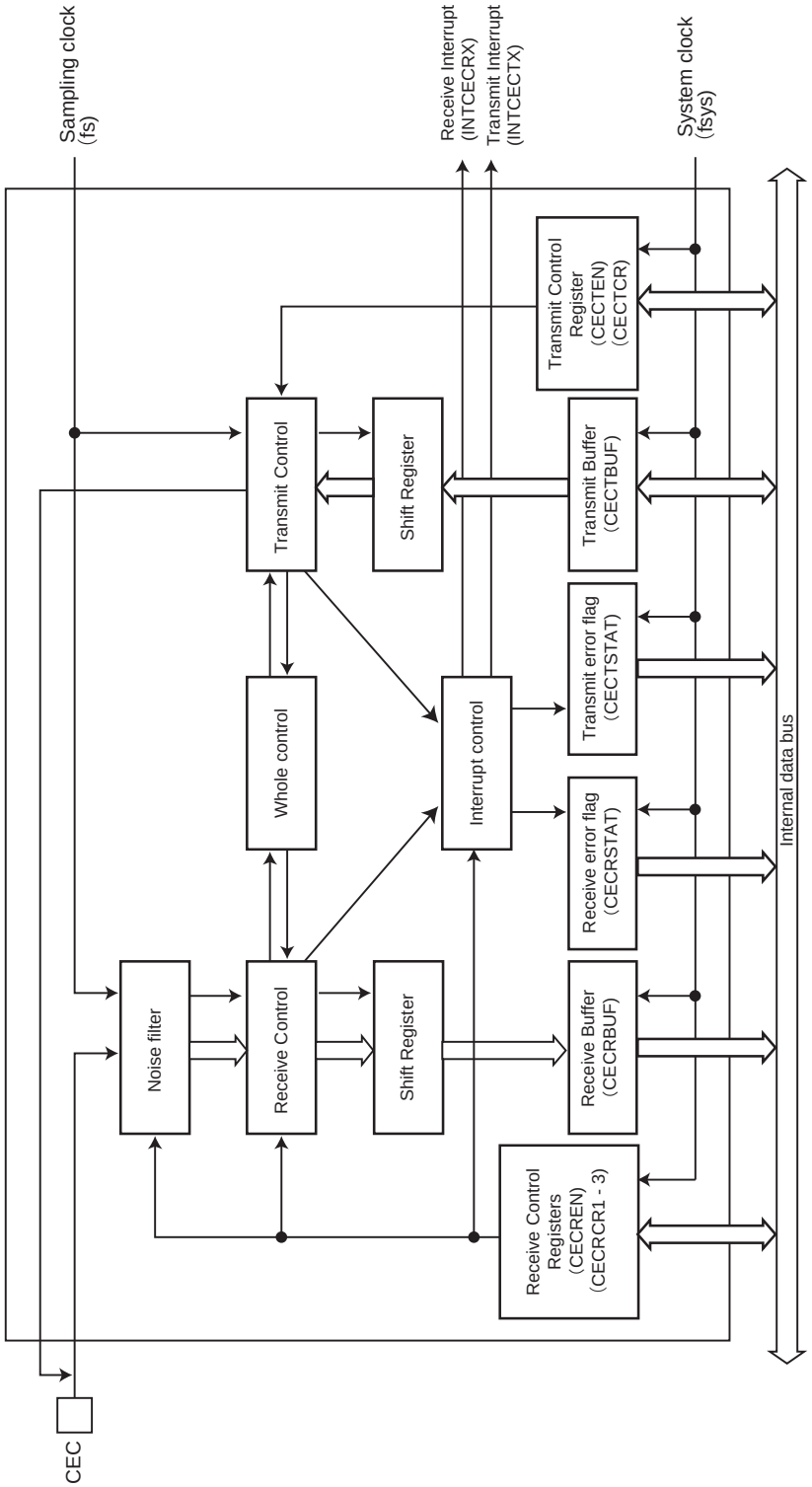


Figure 12-1 Block Diagram of CEC

12.3.10 CECTEN (Transmit Enable 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	-	-	-	-	-	-	CECTRANS	CECTEN
After reset	0	0	0	0	0	0	0	Undefined

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	CECTRANS	R	Transmission state 0: not in progress 1:in progress Indicates whether the transmission is in progress or not. It indicates "1" upon starting the transmission of the start bit. It indicates "0" if transmission is completed or an interrupt is generated. Writing to this bit is ignored.
0	CECTEN	W	Transmission control 0: Disable 1: Enable Controls the CEC transmission. Writing this bit enables or disables the transmission. Writing "1" to this bit initiates the transmission. This bit is automatically cleared by a transmit completion interrupt or an error interrupt.

Note 1: Set <CECTEN> after setting the CECTBUF and CECTCR register.

Note 2: Stop transmission and reception before changing the settings or enabling the transmission and reception.

14. Analog/Digital Converter (ADC)

14.1 Outline

A 10-bit, sequential-conversion analog/digital converter (AD converter) is built into the TMPM330FDFG/FYFG/FWFG.

This AD converter is equipped with 12 analog input channels.

These 12 analog input channels (pins AIN0 through AIN11) are also used as input/output ports.

Note 1: **To assure conversion accuracy, the specified value must be set to the ADCBAS register.**

Note 2: **If it is necessary to reduce a power current by operating the TMPM330FDFG/FYFG/FWFG in IDLE or STOP mode and if either case shown below is applicable, you must first stop the AD converter and then execute the instruction to put the TMPM330FDFG/FYFG/FWFG into standby mode.**

1. The TMPM330FDFG/FYFG/FWFG must be put into IDLE mode when ADMOD1<I2AD> is "0".

2.The TMPM330FDFG/FYFG/FWFG must be put into STOP mode.

Note 1: **Before starting AD conversion, write "1" to the <VREFON> bit, wait for 3μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS>.**

Note 2: **Set <VREFON> to "0" to go into standby mode upon completion of AD conversion.**

16.3.3 Detailed Description of Control Register

16.3.3.1 RTCSECR (Second column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
bit symbol	-	SE						
After reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7	−	R	Read as 0.
6-0	SE	R/W	Setting digit register of second 000_0000 : 00sec. 001_0000 : 10sec. 010_0000 : 20sec. 000_0001 : 01sec. 001_0001 : 11sec. ・ 000_0010 : 02sec. 001_0010 : 12sec. 011_0000 : 30sec. 000_0011 : 03sec. 001_0011 : 13sec. ・ 000_0100 : 04sec. 001_0100 : 14sec. 100_0000 : 40sec. 000_0101 : 05sec. 001_0101 : 15sec. ・ 000_0110 : 06sec. 001_0110 : 16sec. 101_0000 : 50sec. 000_0111 : 07sec. 001_0111 : 17sec. ・ 000_1000 : 08sec. 001_1000 : 18sec. ・ 000_1001 : 09sec. 001_1001 : 19sec. 101_1001 : 59sec.

Note: **The setting other than listed above is prohibited.**

16.3.3.2 RTCMINR (Minute column register (PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	MI						
After reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7	−	R	Read as 0.
6-0	MI	R/W	Setting digit register of Minutes. 000_0000 : 00min. 001_0000 : 10min. 010_0000 : 20min. 000_0001 : 01min. 001_0001 : 11min. ・ 000_0010 : 02min. 001_0010 : 12min. 011_0000 : 30min. 000_0011 : 03min. 001_0011 : 13min. ・ 000_0100 : 04min. 001_0100 : 14min. 100_0000 : 40min. 000_0101 : 05min. 001_0101 : 15min. ・ 000_0110 : 06min. 001_0110 : 16min. 101_0000 : 50min. 000_0111 : 07min. 001_0111 : 17min. ・ 000_1000 : 08min. 001_1000 : 18min. ・ 000_1001 : 09min. 001_1001 : 19min. 101_1001 : 59min.

Note: **The setting other than listed above is prohibited.**

17.2.2 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the old application.

The condition to switch the modes needs to be set by using the I/O of TMPM330FDFG/FYFG/FWFG in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete/ writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. Be sure not to cause any exceptions including a non-maskable while User Boot Mode.

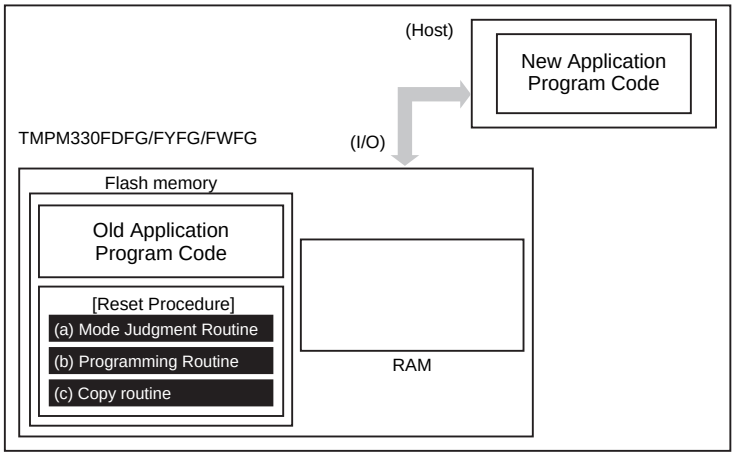
(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to "17.3 On-board Programming of Flash Memory (Rewrite/Erase)".

17.2.2.1 (1-A) Method 1: Storing a Programming Routine in the Flash Memory

(1) Step-1

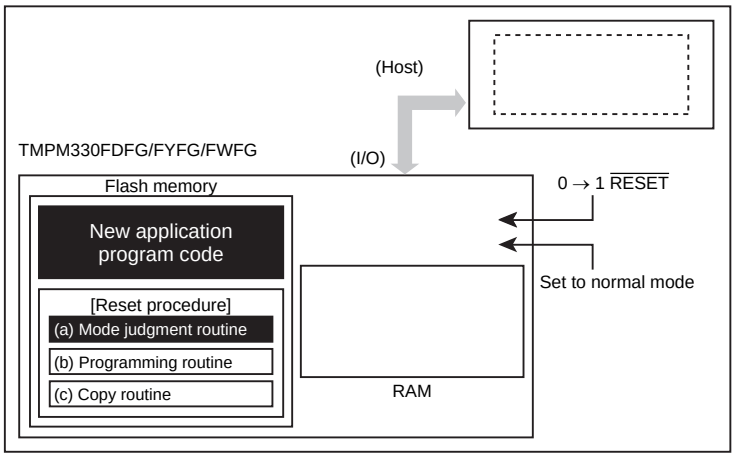
Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM330FDFG/FYFG/FWFG on a printed circuit board, write the following program routines 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) Programming routine:	Code to download new program code from a host controller and re-program the flash memory
(c) Copy routine:	Code to copy the data described in (b) from the TMPM330FDFG/FYFG/FWFG flash memory to either the TMPM330FDFG/FYFG/FWFG on-chip RAM or external memory device.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" to reset the TMPM330FDFG/FYFG/FWFG. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



17.2.9.2 Show Flash Memory SUM

Table 17-7 Transfer Format for the Show Flash Memory SUM Command

	Byte	Data Transferred from the Controller to the TMPM330FDFG/FYFG/FWFG	Baud rate	Data Transferred from the TMPM330FDFG/ FYFG/FWFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode: 0x30	Desired baud rate (Note 1)	−
	2 byte	−		ACK for the serial operation mode byte ・For UART mode -Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode -Normal acknowledge : 0x30
	3 byte	Command code (0x20)		−
	4 byte	−		ACK for the command code byte (Note 2) -Normal acknowledge : 0x20 -Negative acknowledge : 0xX1 -Communication error : 0xX8
	5 byte	−		SUM (upper byte)
	6 byte	−		SUM (lower byte)
	7 byte	−		Checksum value for bytes 5 and 6
	8 byte	(Wait for the next command code.)		−

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**