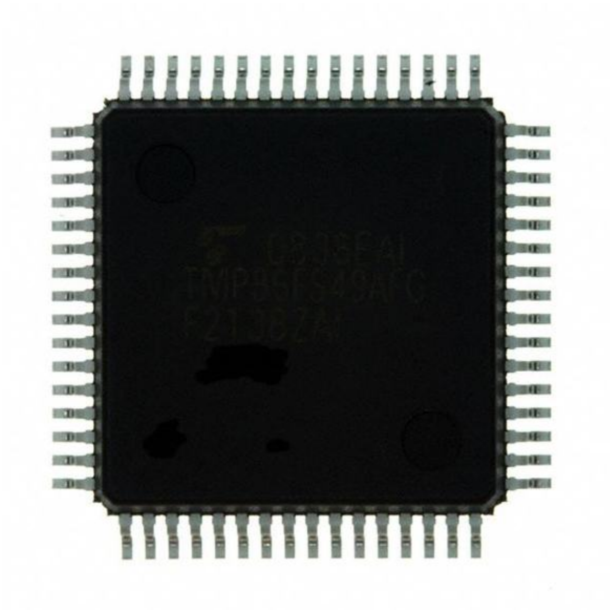
Toshiba Semiconductor and Storage - TMP86FS49BFG(CZHZ) Datasheet



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	870/C
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, SIO, UART/USART
Peripherals	LED, PWM, WDT
Number of I/O	56
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-BQFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmp86fs49bfg-czhz

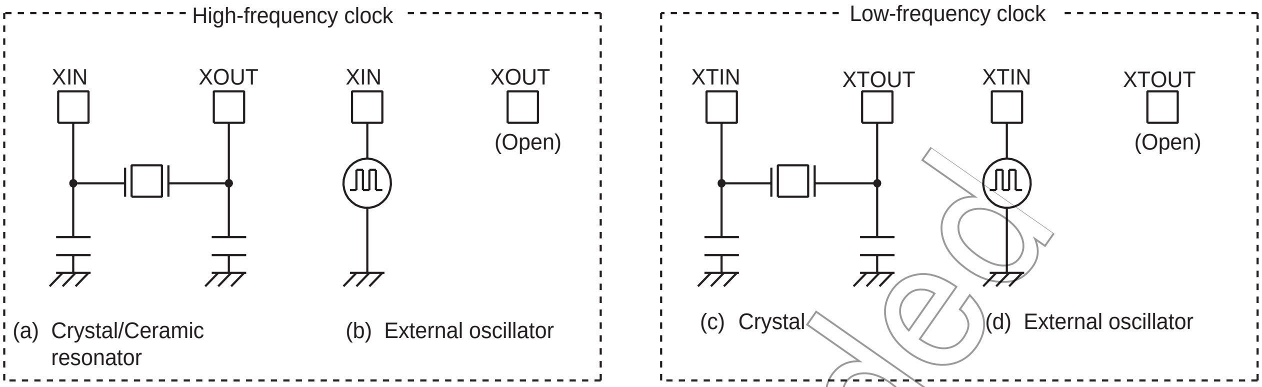


Figure 2-3 Examples of Resonator Connection

Note: The function to monitor the basic clock directly at external is not provided for hardware, however, with disabling all interrupts and watchdog timers, the oscillation frequency can be adjusted by monitoring the pulse which the fixed frequency is outputted to the port by the program.
The system to require the adjustment of the oscillation frequency should create the program for the adjustment in advance.

P1DR (0001H) R/W (Initial value: 0000 0000)

	7	6	5	4	3	2	1	0
P1DR (0001H) R/W	P17 TC6 $\overline{PWM6}$ $\overline{PDO6}$ $\overline{PPG6}$	P16 TC5 $\overline{PWM5}$ $\overline{PDO5}$	P15 TC2 INT3	P14 TC4 $\overline{PWM4}$ $\overline{PDO4}$ $\overline{PPG4}$	P13 TC3 $\overline{PWM3}$ $\overline{PDO3}$	P12 $\overline{PPG}$	P11 $\overline{DVO}$	P10 TC1

P1CR (0009H) (Initial value: 0000 0000)

	7	6	5	4	3	2	1	0
P1CR (0009H)								

P1CR	I/O control for port P1 (Specified for each bit)	0: Input mode 1: Output mode	R/W

8.2 TimerCounter Control

The TimerCounter 1 is controlled by the TimerCounter 1 control register (TC1CR) and two 16-bit timer registers (TC1DRA and TC1DRB).

Timer Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC1DRA (0011H, 0010H)	TC1DRAH (0011H)								TC1DRAL (0010H)							
	(Initial value: 1111 1111 1111 1111)								Read/Write							
TC1DRB (0013H, 0012H)	TC1DRBH (0013H)								TC1DRBL (0012H)							
	(Initial value: 1111 1111 1111 1111)								Read/Write (Write enabled only in the PPG output mode)							

TimerCounter 1 Control Register

	7	6	5	4	3	2	1	0	
TC1CR (0026H)	TFF1	ACAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M		Read/Write (Initial value: 0000 0000)

TFF1	Timer F/F1 control	0: Clear			1: Set				R/W
ACAP1	Auto capture control	0:Auto-capture disable			1:Auto-capture enable				R/W
MCAP1	Pulse width measurement mode control	0:Double edge capture			1:Single edge capture				R/W
METT1	External trigger timer mode control	0:Trigger start			1:Trigger start and stop				R/W
MPPG1	PPG output control	0:Continuous pulse generation			1:One-shot				R/W

			Timer	Extrigger	Event	Window	Pulse	PPG	
TC1S	TC1 start control	00: Stop and counter clear	O	O	O	O	O	O	R/W
		01: Command start	O	–	–	–	–	O	
		10: Rising edge start (Ex-trigger/Pulse/PPG) Rising edge count (Event) Positive logic count (Window)	–	O	O	O	O	O	
		11: Falling edge start (Ex-trigger/Pulse/PPG) Falling edge count (Event) Negative logic count (Window)	–	O	O	O	O	O	

			NORMAL1/2, IDLE1/2 mode		Divider	SLOW, SLEEP mode	
			DV7CK = 0	DV7CK = 1			
TC1CK	TC1 source clock select [Hz]	00	$fc/2^{11}$	$fs/2^3$	DV9	$fs/2^3$	R/W
		01	$fc/2^7$	$fc/2^7$	DV5	–	
		10	$fc/2^3$	$fc/2^3$	DV1	–	
		11	External clock (TC1 pin input)				

TC1M	TC1 operating mode select	00: Timer/external trigger timer/event counter mode 01: Window mode 10: Pulse width measurement mode 11: PPG (Programmable pulse generate) output mode	R/W

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]

Note 2: The timer register consists of two shift registers. A value set in the timer register becomes valid at the rising edge of the first source clock pulse that occurs after the upper byte (TC1DRAH and TC1DRBH) is written. Therefore, write the lower byte and the upper byte in this order (it is recommended to write the register with a 16-bit access instruction). Writing only the lower byte (TC1DRAL and TC1DRBL) does not enable the setting of the timer register.

Note 3: To set the mode, source clock, PPG output control and timer F/F control, write to TC1CR during TC1S=00. Set the timer F/F1 control until the first timer start after setting the PPG mode.

8.3 Function

TimerCounter 1 has six types of operating modes: timer, external trigger timer, event counter, window, pulse width measurement, programmable pulse generator output modes.

8.3.1 Timer mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register 1A (TC1DRA) value is detected, an INTTC1 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting. Setting TC1CR<ACAP1> to “1” captures the up-counter value into the timer register 1B (TC1DRB) with the auto-capture function. Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition. Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

Table 8-1 Internal Source Clock for TimerCounter 1 (Example: fc = 16 MHz, fs = 32.768 kHz)

TC1CK	NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode	
	DV7CK = 0		DV7CK = 1			
	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]
00	128	8.39	244.14	16.0	244.14	16.0
01	8.0	0.524	8.0	0.524	–	–
10	0.5	32.77 m	0.5	32.77 m	–	–

Example 1 :Setting the timer mode with source clock fc/2^{11} [Hz] and generating an interrupt 1 second later (fc = 16 MHz, TBTCR<DV7CK> = “0”)

```
LDW   (TC1DRA), 1E84H   ; Sets the timer register (1 s ÷ 2^11/fc = 1E84H)
DI                      ; IMF= “0”
SET   (EIRL). 5         ; Enables INTTC1
EI                      ; IMF= “1”
LD    (TC1CR), 00000000B ; Selects the source clock and mode
LD    (TC1CR), 00010000B ; Starts TC1
```

Example 2 :Auto-capture

```
LD    (TC1CR), 01010000B ; ACAP1 ← 1
:     :
LD    WA, (TC1DRB)       ; Reads the capture value
```

Note: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

9.2 Control

The timer/counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TC2DR).

TC2DR (0025H, 0024H)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TC2DRH (0025H)								TC2DRL (0024H)							

(Initial value: 1111 1111 1111 1111) R/W

TC2CR (0023H)	7	6	5	4	3	2	1	0
			TC2S	TC2CK				TC2M

(Initial value: **00 00*0)

<table>
<tr><td>TC2S</td><td>TC2 start control</td><td colspan="7">0:Stop and counter clear
1:Start</td><td>R/W</td></tr>
<tr><td rowspan="10">TC2CK</td><td rowspan="10">TC2 source clock select
Unit : [Hz]</td><td rowspan="2"></td><td colspan="2">NORMAL1/2, IDLE1/2 mode</td><td rowspan="2">Divider</td><td rowspan="2">SLOW1/2 mode</td><td rowspan="2">SLEEP1/2 mode</td><td rowspan="10">R/W</td></tr>
<tr><td>DV7CK = 0</td><td>DV7CK = 1</td></tr>
<tr><td>000</td><td>$fc/2^{23}$</td><td>$fs/2^{15}$</td><td>DV21</td><td>$fs/2^{15}$</td><td>$fs/2^{15}$</td></tr>
<tr><td>001</td><td>$fc/2^{13}$</td><td>$fs/2^{5}$</td><td>DV11</td><td>$fs/2^{5}$</td><td>$fs/2^{5}$</td></tr>
<tr><td>010</td><td>$fc/2^{8}$</td><td>$fc/2^{8}$</td><td>DV6</td><td>–</td><td>–</td></tr>
<tr><td>011</td><td>$fc/2^{3}$</td><td>$fc/2^{3}$</td><td>DV1</td><td>–</td><td>–</td></tr>
<tr><td>100</td><td>–</td><td>–</td><td>–</td><td>fc (Note7)</td><td>–</td></tr>
<tr><td>101</td><td>fs</td><td>fs</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>110</td><td colspan="5">Reserved</td></tr>
<tr><td>111</td><td colspan="5">External clock (TC2 pin input)</td></tr>
<tr><td>TC2M</td><td>TC2 operating mode select</td><td colspan="7">0:Timer/event counter mode
1:Window mode</td><td>R/W</td></tr>
</table>

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 2: When writing to the Timer Register 2 (TC2DR), always write to the lower side (TC2DRL) and then the upper side (TC2DRH) in that order. Writing to only the lower side (TC2DRL) or the upper side (TC2DRH) has no effect.

Note 3: The timer register 2 (TC2DR) uses the value previously set in it for coincidence detection until data is written to the upper side (TC2DRH) after writing data to the lower side (TC2DRL).

Note 4: Set the mode and source clock when the TC2 stops (TC2S = 0).

Note 5: Values to be loaded to the timer register must satisfy the following condition.
TC2DR > 1 ($TC2DR_{15}$ to $TC2DR_{11}$ > 1 at warm up)

Note 6: If a read instruction is executed for TC2CR, read data of bit 7, 6 and 1 are unstable.

Note 7: The high-frequency clock (fc) canbe selected only when the time mode at SLOW2 mode is selected.

Note 8: On entering STOP mode, the TC2 start control (TC2S) is cleared to "0" automatically. So, the timer stops. Once the STOP mode has been released, to start using the timer counter, set TC2S again.

9.3 Function

The timer/counter 2 has three operating modes: timer, event counter and window modes.

And if fc or fs is selected as the source clock in timer mode, when switching the timer mode from SLOW1 to NORMAL2, the timer/counter2 can generate warm-up time until the oscillator is stable.

9.3.1 Timer mode

In this mode, the internal clock is used for counting up. The contents of TC2DR are compared with the contents of up counter. If a match is found, a timer/counter 2 interrupt (INTTC2) is generated, and the counter is cleared. Counting up is resumed after the counter is cleared.

When fc is selected for source clock at SLOW2 mode, lower 11-bits of TC2DR are ignored and generated a interrupt by matching upper 5-bits only. Though, in this situation, it is necessary to set TC2DRH only.

Table 9-1 Source Clock (Internal clock) for Timer/Counter2 (at fc = 16 MHz, DV7CK=0)

TC2CK	NORMAL1/2, IDLE1/2 mode				SLOW1/2 mode		SLEEP1/2 mode	
	DV7CK = 0		DV7CK = 1					
	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting
000	524.29 [ms]	9.54 [h]	1 [s]	18.2 [h]	1 [s]	18.2 [h]	1 [s]	18.2 [h]
001	512.0 [ms]	33.55 [s]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]
010	16.0 [ms]	1.05 [s]	16.0 [ms]	1.05 [s]	–	–	–	–
011	0.5 [ms]	32.77 [ms]	0.5 [ms]	32.77 [ms]	–	–	–	–
100	–	–	–	–	62.5 [ns]	–	–	–
101	30.52 [ms]	2 [s]	30.52 [ms]	2 [s]	–	–	–	–

Note: When fc is selected as the source clock in timer mode, it is used at warm-up for switching from SLOW1 mode to NORMAL2 mode.

Example :Sets the timer mode with source clock fc/2^3 [Hz] and generates an interrupt every 25 ms (at fc = 16 MHz)

```
LDW   (TC2DR), 061AH        ; Sets TC2DR (25 ms ³ 2^8/fc = 061AH)
DI                          ; IMF= "0"
SET   (EIRE). 6             ; Enables INTTC2 interrupt
EI                          ; IMF= "1"
LD    (TC2CR), 00001000B    ; Source clock / mode select
LD    (TC2CR), 00101000B    ; Starts Timer
```

10. 8-Bit TimerCounter (TC3, TC4)

10.1 Configuration

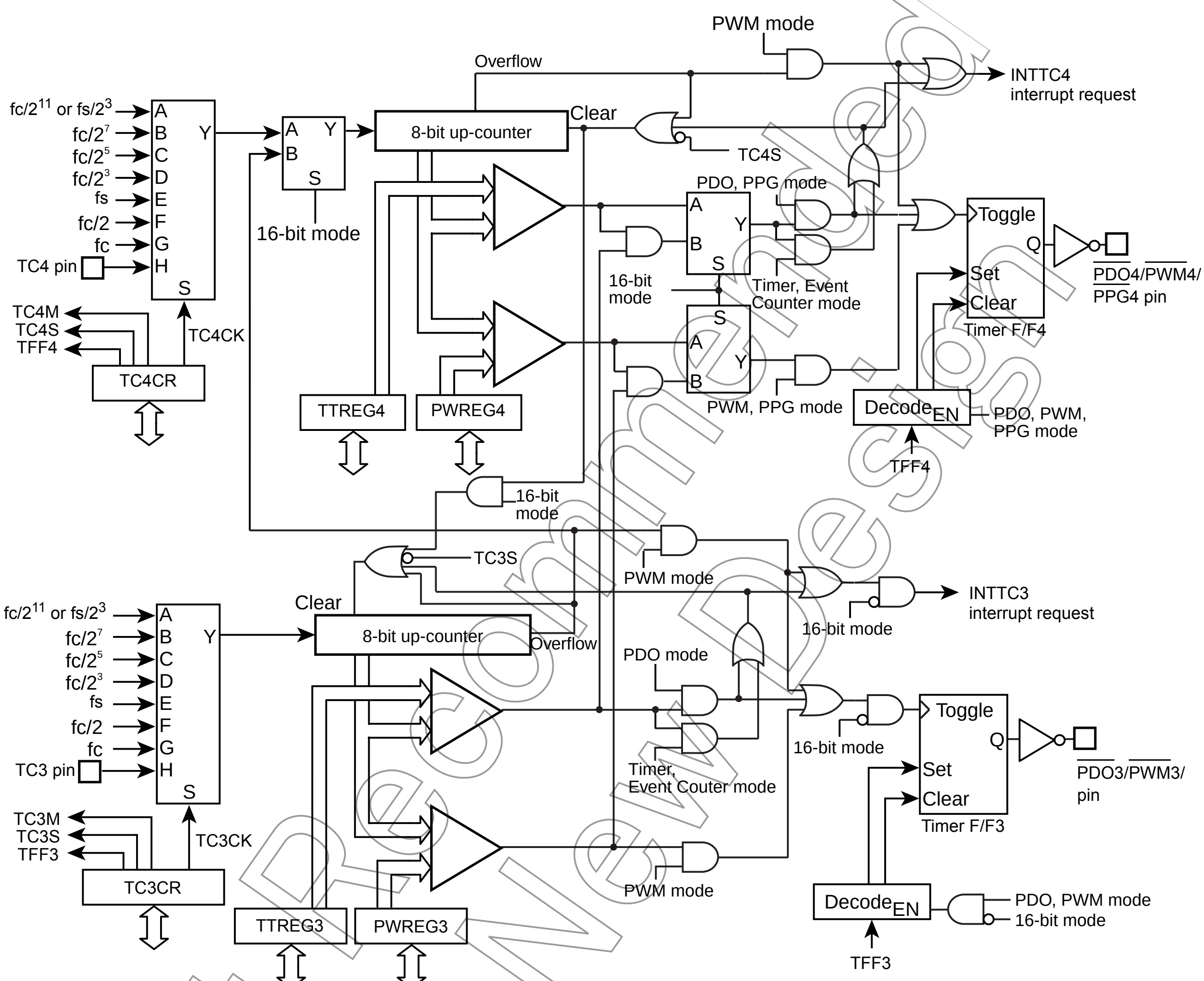


Figure 10-1 8-Bit TimerCounter 3, 4

Note 7: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-3.

Note 8: The operating clock fc in the SLOW or SLEEP mode can be used only as the high-frequency warm-up mode.

The TimerCounter 4 is controlled by the TimerCounter 4 control register (TC4CR) and two 8-bit timer registers (TTREG4 and PWREG4).

TimerCounter 4 Timer Register

TTREG4 (0015H) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

PWREG4 (0019H) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG4) setting while the timer is running.
Note 2: Do not change the timer register (PWREG4) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

TimerCounter 4 Control Register

TC4CR (0028H)	7	6	5	4	3	2	1	0	
	TFF4	TC4CK			TC4S	TC4M			(Initial value: 0000 0000)

TFF4	Timer F/F4 control	0: Clear 1: Set				R/W
TC4CK	Operating clock selection [Hz]		NORMAL1/2, IDLE1/2 mode		SLOW1/2 SLEEP1/2 mode	R/W
			DV7CK = 0	DV7CK = 1		
		000	fc/2^{11}	fs/2^3	fs/2^3	
		001	fc/2^7	fc/2^7	–	
		010	fc/2^5	fc/2^5	–	
		011	fc/2^3	fc/2^3	–	
		100	fs	fs	fs	
		101	fc/2	fc/2	–	
		110	fc	fc	–	
		111	TC4 pin input			
TC4S	TC4 start control	0: Operation stop and counter clear 1: Operation start				R/W
TC4M	TC4M operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: Reserved 100: 16-bit timer/event counter mode 101: Warm-up counter mode 110: 16-bit pulse width modulation (PWM) output mode 111: 16-bit PPG mode				R/W

Note 1: fc: High-frequency clock [Hz] fs: Low-frequency clock [Hz]
Note 2: Do not change the TC4M, TC4CK and TFF4 settings while the timer is running.
Note 3: To stop the timer operation (TC4S= 1 → 0), do not change the TC4M, TC4CK and TFF4 settings.
To start the timer operation (TC4S= 0 → 1), TC4M, TC4CK and TFF4 can be programmed.
Note 4: When TC4M= 1** (upper byte in the 16-bit mode), the source clock becomes the TC3 overflow signal regardless of the TC4CK setting.
Note 5: To use the TimerCounter in the 16-bit mode, select the operating mode by programming TC4M, where TC3CR<TC3M> must be set to 011.

Example :Generating 1024 Hz pulse using TC4 (fc = 16.0 MHz)

	Setting port	
LD	(TTREG4), 3DH	: 1/1024÷2^7/fc÷2 = 3DH
LD	(TC4CR), 00010001B	: Sets the operating clock to fc/2^7, and 8-bit PDO mode.
LD	(TC4CR), 00011001B	: Starts TC4.

Note 1: In the programmable divider output mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the programmable divider output mode, the new value programmed in TTREGj is in effect immediately after programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PDO output, the $\overline{PDOj}$ pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> setting upon stopping of the timer.
Example: Fixing the $\overline{PDOj}$ pin to the high level when the TimerCounter is stopped
CLR (TCjCR).3: Stops the timer.
CLR (TCjCR).7: Sets the $\overline{PDOj}$ pin to the high level.

Note 3: j = 3, 4

11.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC5, 6)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 8 bits of resolution. The up-counter counts up using the internal clock.

When a match between the up-counter and the PWREGj value is detected, the logic level output from the timer F/Fj is switched to the opposite state. The counter continues counting. The logic level output from the timer F/Fj is switched to the opposite state again by the up-counter overflow, and the counter is cleared. The INTTCj interrupt request is generated at this time.

Since the initial value can be set to the timer F/Fj by TCjCR<TFFj>, positive and negative pulses can be generated. Upon reset, the timer F/Fj is cleared to 0.

(The logic level output from the $\overline{\text{PWMj}}$ pin is the opposite to the timer F/Fj logic level.)

Since PWREGj in the PWM mode is serially connected to the shift register, the value set to PWREGj can be changed while the timer is running. The value set to PWREGj during a run of the timer is shifted by the INTTCj interrupt request and loaded into PWREGj. While the timer is stopped, the value is shifted immediately after the programming of PWREGj. If executing the read instruction to PWREGj during PWM output, the value in the shift register is read, but not the value set in PWREGj. Therefore, after writing to PWREGj, the reading data of PWREGj is previous value until INTTCj is generated.

For the pin used for PWM output, the output latch of the I/O port must be set to 1.

Note 1: In the PWM mode, program the timer register PWREGj immediately after the INTTCj interrupt request is generated (normally in the INTTCj interrupt service routine.) If the programming of PWREGj and the interrupt request occur at the same time, an unstable value is shifted, that may result in generation of the pulse different from the programmed value until the next INTTCj interrupt request is generated.

Note 2: When the timer is stopped during PWM output, the $\overline{\text{PWMj}}$ pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> upon stopping of the timer.
Example: Fixing the $\overline{\text{PWMj}}$ pin to the high level when the TimerCounter is stopped
CLR (TCjCR).3: Stops the timer.
CLR (TCjCR).7: Sets the $\overline{\text{PWMj}}$ pin to the high level.

Note 3: To enter the STOP mode during PWM output, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping the timer when fc, fc/2 or fs is selected as the source clock, a pulse is output from the $\overline{\text{PWMj}}$ pin during the warm-up period time after exiting the STOP mode.

Note 4: j = 5, 6

Table 11-5 PWM Output Mode

Source Clock			Resolution		Repeated Cycle	
NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	128 μs	244.14 μs	32.8 ms	62.5 ms
$fc/2^7$	$fc/2^7$	–	8 μs	–	2.05 ms	–
$fc/2^5$	$fc/2^5$	–	2 μs	–	512 μs	–
$fc/2^3$	$fc/2^3$	–	500 ns	–	128 μs	–
fs	fs	fs	30.5 μs	30.5 μs	7.81 ms	7.81 ms
fc/2	fc/2	–	125 ns	–	32 μs	–
fc	fc	–	62.5 ns	–	16 μs	–

13. Asynchronous Serial interface (UART2)

13.1 Configuration

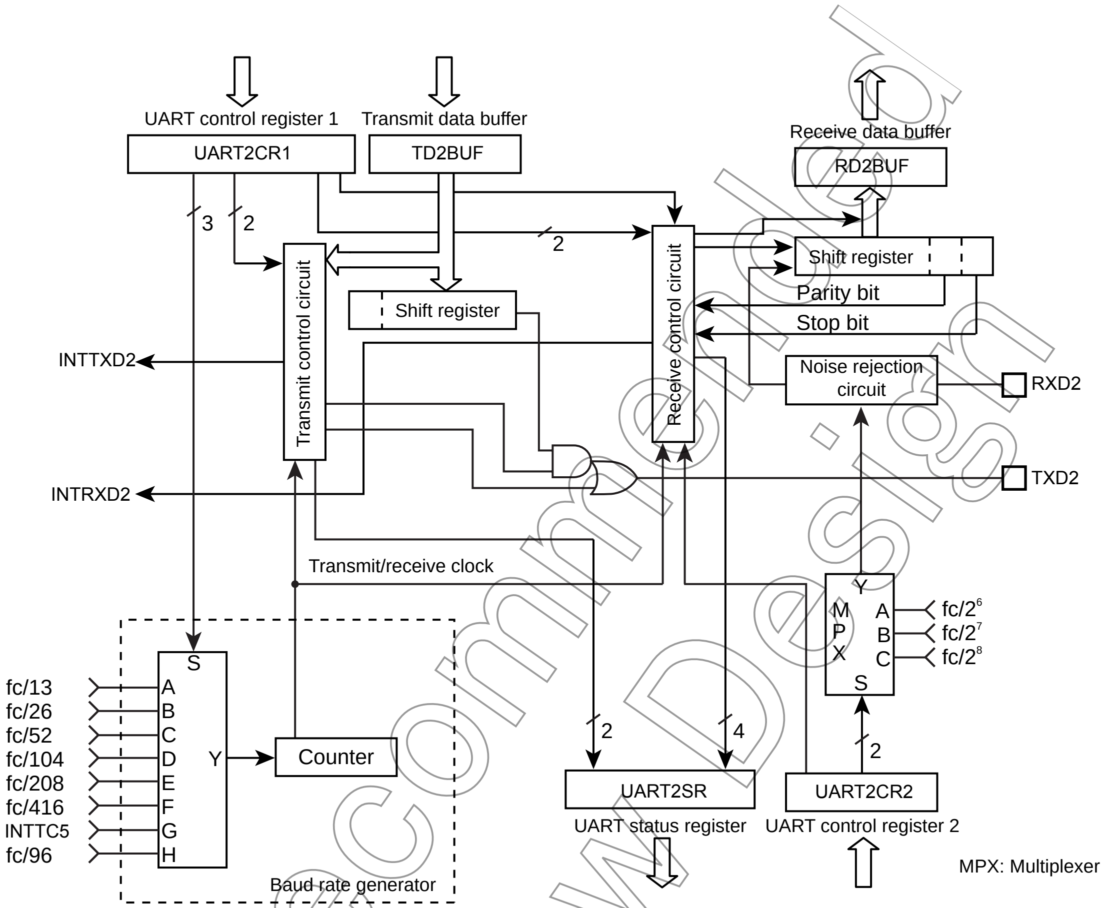


Figure 13-1 UART2 (Asynchronous Serial Interface)

14.2 Control

The SIO is controlled using the serial interface control register (SIO1CR). The operating status of the serial interface can be inspected by reading the status register (SIO1CR).

Serial Interface Control Register

SIO1CR (0020H)	7	6	5	4	3	2	1	0	
	SIOS	SIOINH	SIOM		SIODIR	SCK			(Initial value: 0000 0000)

SIOS	Specify start/stop of transfer	0: Stop 1: Start				R/W
SIOINH	Forcibly stops transfer (Note 1)	0: – 1: Forcibly stop (Automatically cleared to "0" after stopping)				
SIOM	Selects transfer mode	00: Transmit mode 01: Receive mode 10: Transmit/receive mode 11: Reserved				
SIODIR	Selects direction of transfer	0: MSB (Transfer beginning with bit7) 1: LSB (Transfer beginning with bit0)				
SCK	Selects serial clock		NORMAL1/2 or IDLE1/2 modes		SLOW/SLEEP mode	
			TBTCR <DV7CK> = "0"	TBTCR <DV7CK> = "1"		
		000	$fc/2^{12}$	$fs/2^4$	$fs/2^4$	
		001	$fc/2^8$	$fc/2^8$	Reserved	
		010	$fc/2^7$	$fc/2^7$	Reserved	
		011	$fc/2^6$	$fc/2^6$	Reserved	
		100	$fc/2^5$	$fc/2^5$	Reserved	
		101	$fc/2^4$	$fc/2^4$	Reserved	
		110	$fc/2^3$	$fc/2^3$	Reserved	
		111	External clock (Input from $\overline{SCK1}$ pin)			

Note 1: When SIO1CR<SIOINH> is set to "1", SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

Note 2: Transfer mode, direction of transfer and serial clock must be select during the transfer is stopping (when SIO1SR<SIOF> "0").

Note 3: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

(2) During the transmit/receive operation

When data is written to SIO1TDB, SIO1SR<TXF> is cleared to “0” and when a data is read from SIO1RDB, SIO1SR<RXF> is cleared to “0”.

In internal clock operation, in case of the condition described below, the serial clock stops to “H” level by an automatic-wait function when all of the bit set in the data has been transmitted.

- Next transmit data is not written to SIO1TDB after reading a received data from SIO1RDB.
- Received data is not read from SIO1RDB after writing a next transmit data to SIO1TDB.
- Neither SIO1TDB nor SIO1RDB is accessed after transmission.

The automatic wait function is released by writing the next transmit data to SIO1TDB after reading the received data from SIO1RDB, or reading the received data from SIO1RDB after writing the next data to SIO1TDB.
Then, transmit/receive operation is restarted after maximum 1 cycle of serial clock.
In external clock operation, reading the received data from SIO1RDB and writing the next data to SIO1TDB must be finished before the shift operation of the next data begins.
If the transmit data is not written to SIO1TDB after SIO1SR<TXF> is set to “1”, transmit error occurs immediately after shift operation is started. When the transmit error occurred, SIO1SR<TXERR> is set to “1”.
If received data is not read out from SIO1RDB before next shift operation starts after setting SIO1SR<RXF> to “1”, receive error occurs immediately after shift operation is finished. When the receive error has occurred, SIO1SR<RXERR> is set to “1”.

(3) Stopping the transmit/receive operation

There are two ways for stopping the transmit/receive operation.

- The way of clearing SIO1CR<SIOS>.
  When SIO1CR<SIOS> is cleared to “0”, transmit/receive operation is stopped after all transfer of the data is finished. When transmit/receive operation is finished, SIO1SR<SIOF> is cleared to “0” and SO1 pin is kept in high level.
  In external clock operation, SIO1CR<SIOS> must be cleared to “0” before SIO1SR<SEF> is set to “1” by beginning next transfer.
- The way of setting SIO1CR<SIOINH>.
  Transmit/receive operation is stopped immediately after SIO1CR<SIOINH> is set to “1”. In this case, SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

16. Serial Bus Interface(I^2C Bus) Ver.-D (SBI)

The TMP86FS49BFG has a serial bus interface which employs an I^2C bus.

The serial interface is connected to an external devices through SDA and SCL.

The serial bus interface pins are also used as the port. When used as serial bus interface pins, set the output latches of these pins to "1". When not used as serial bus interface pins, the port is used as a normal I/O port.

Note 1: The serial bus interface can be used only in NORMAL1/2 and IDLE1/2 mode. It can not be used in IDLE0, SLOW1/2 and SLEEP0/1/2 mode.

Note 2: The serial bus interface can be used only in the Standard mode of I^2C. The fast mode and the high-speed mode can not be used.

Note 3: Please refer to the I/O port section about the detail of setting port.

16.1 Configuration

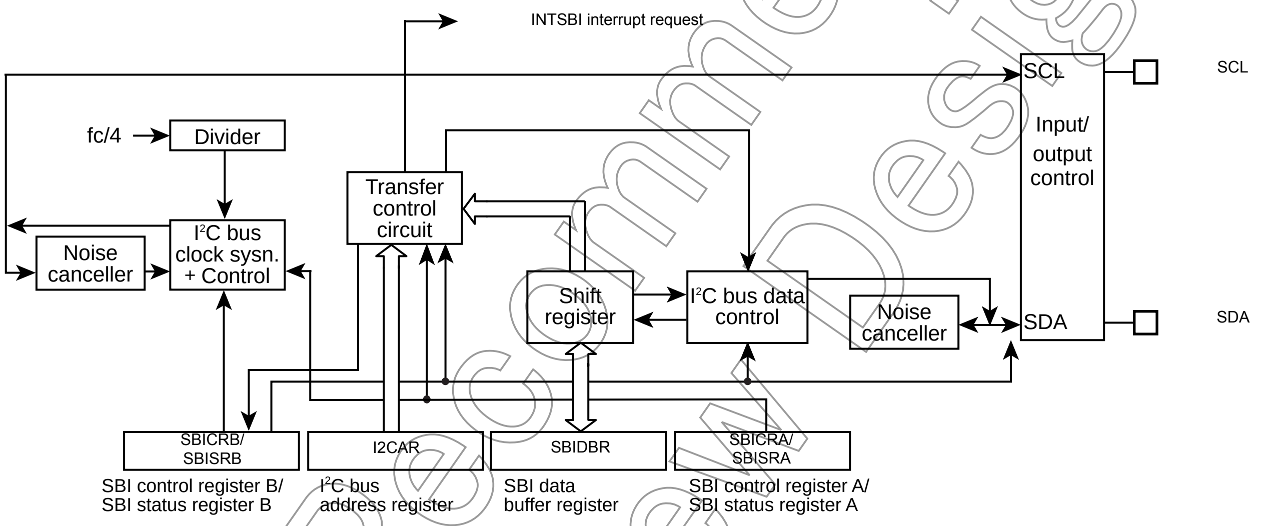


Figure 16-1 Serial Bus Interface (SBI)

16.2 Control

The following registers are used for control the serial bus interface and monitor the operation status.

- Serial bus interface control register A (SBICRA)
- Serial bus interface control register B (SBICRB)
- Serial bus interface data buffer register (SBIDBR)
- I^2C bus address register (I2CAR)
- Serial bus interface status register A (SBISRA)
- Serial bus interface status register B (SBISRB)

16.3 Software Reset

A serial bus interface circuit has a software reset function, when a serial bus interface circuit is locked by an external noise, etc.

To reset the serial bus interface circuit, write “10”, “01” into the SWRST (Bit1, 0 in SBICRB).

And a status of software reset canbe read from SWRMON (Bit0 in SBISRA).

In the slave mode, the conditions of generating INTSBI interrupt request are follows:

- At the end of acknowledge signal when the received slave address matches to the value set by the I2CAR
- At the end of acknowledge signal when a “GENERAL CALL” is received
- At the end of transferring or receiving after matching of slave address or receiving of “GENERAL CALL”

When a serial bus interface interrupt request occurs, the PIN (Bit4 in SBISRB) is cleared to “0”. During the time that the PIN is “0”, the SCL pin is pulled-down to low level.

Either writing data to SBIDBR or reading data from the SBIDBR sets the PIN to “1”.

The time from the PIN being set to “1” until the SCL pin is released takes t_{LOW}.

Although the PIN (Bit4 in SBICRB) can be set to “1” by the softrware, the PIN can not be cleared to “0” by the softrware.

Note: When the arbitration lost occurs, if the slave address sent from the other master devices is not match, the INTSBI interrupt request is generated. But the PIN is not cleared.

16.5.9 Setting of I^2C bus mode

The SBIM (Bit3 and 2 in SBICRB) is used to set I^2C bus mode.

Set the SBIM to “10” in order to set I^2C bus mode. Before setting of I^2C bus mode, confirm serial bus interface pins in a high level, and then, write “10” to SBIM. And switch a port mode after confirming that a bus is free.

16.5.10 Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I^2C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master 1 and Master 2 output the same data until point “a”. After that, when Master 1 outputs “1” and Master 2 outputs “0”, since the SDA line of a bus is wired AND, the SDA line is pulled-down to the low level by Master 2. When the SCL line of a bus is pulled-up at point “b”, the slave device reads data on the SDA line, that is data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master 1 is called “arbitration lost”. A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

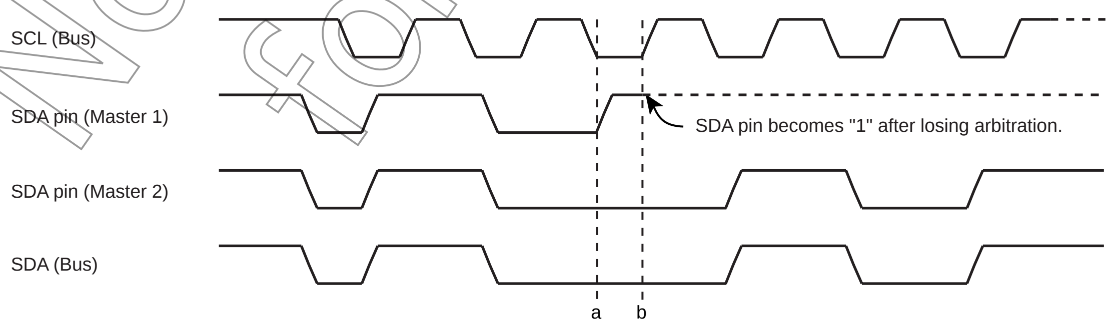


Figure 16-7 Arbitration Lost

The serial bus interface circuit compares levels of a SDA line of a bus with its SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and the AL (Bit3 in SBISRB) is set to “1”.

Example :After selecting the conversion time 19.5 µs at 16 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value, store the lower 2 bits in address 0009EH nd store the upper 8 bits in address 0009FH in RAM. The operation mode is software start mode.

```
          : (port setting)   :                       ;Set port register approrriately before setting AD
                                                      converter registers.
          :                  :                       (Refer to section I/O port in details)
          LD                 (ADCCR1) ,  00100011B   ; Select AIN3
          LD                 (ADCCR2) ,  11011000B   ;Select conversion time(312/fc) and operation
                                                      mode

          SET                (ADCCR1) . 7            ; ADRS = 1(AD conversion start)
SLOOP :   TEST               (ADCDR2) . 5            ; EOCF= 1 ?
          JRS                T, SLOOP

          LD                 A , (ADCDR2)            ; Read result data
          LD                 (9EH) , A
          LD                 A , (ADCDR1)            ; Read  result data
          LD                 (9FH), A
```

17.4 STOP/SLOW Modes during AD Conversion

When standby mode (STOP or SLOW mode) is entered forcibly during AD conversion, the AD convert operation is suspended and the AD converter is initialized (ADCCR1 and ADCCR2 are initialized to initial value). Also, the conversion result is indeterminate. (Conversion results up to the previous operation are cleared, so be sure to read the conversion results before entering standby mode (STOP or SLOW mode).) When restored from standby mode (STOP or SLOW mode), AD conversion is not automatically restarted, so it is necessary to restart AD conversion. Note that since the analog reference voltage is automatically disconnected, there is no possibility of current flowing into the analog reference voltage.

20.3 Serial PROM Mode Setting

20.3.1 Serial PROM Mode Control Pins

To execute on-board programming, activate the serial PROM mode. Table 20-2 shows pin setting to activate the serial PROM mode.

Table 20-2 Serial PROM Mode Setting

Pin	Setting
TEST pin	High
BOOT/RXD1 pin	High
$\overline{RESET}$ pin	

Note: The BOOT pin is shared with the UART communication pin (RXD1 pin) in the serial PROM mode. This pin is used as UART communication pin after activating serial PROM mode

20.3.2 Pin Function

In the serial PROM mode, TXD1 (P02) and RXD1 (P01) are used as a serial interface pin.

Table 20-3 Pin Function in the Serial PROM Mode

Pin Name (Serial PROM Mode)	Input/ Output	Function		Pin Name (MCU Mode)
TXD1	Output	Serial data output	(Note 1)	P02
BOOT/RXD1	Input/Input	Serial PROM mode control/Serial data input	(Note 1)	P01
$\overline{RESET}$	Input	Serial PROM mode control	(Note 1)	$\overline{RESET}$
TEST	Input	Fixed to high		TEST
VDD, AVDD	Power supply	4.5 to 5.5 V		
VSS	Power supply	0 V		
VAREF	Power supply	Leave open or apply input reference voltage.		
I/O ports except P02, P01	I/O	These ports are in the high-impedance state in the serial PROM mode.		
XIN	Input	Self-oscillate with an oscillator.		(Note 2)
XOUT	Output			

Note 1: During on-board programming with other parts mounted on a user board, be careful no to affect these communication control pins.

Note 2: Operating range of high frequency in serial PROM mode is 2 MHz to 16 MHz.

23. Package Dimensions

QFP64-P-1414-0.80A Rev 01

Unit: mm