E. Kentoshiba Semiconductor and Storage - <u>TMP86FS49BUG(C,JZ) Datasheet</u>



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	870/C
Core Size	8-Bit
Speed	16MHz
Connectivity	I ² C, 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-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmp86fs49bug-c-jz

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Caution in Setting the UART Noise Rejection Time

When UART is used, settings of RXDNC are limited depending on the transfer clock specified by BRG. The combination "O" is available but please do not select the combination "-".

 $\Box \uparrow \land$

The transfer clock generated by timer/counter interrupt is calculated by the following equation :

Transfer clock [Hz] = Timer/counter source clock [Hz] ÷ TTREG set value

			\langle	$\langle (/) \rangle$			
BRG setting		RXDNC setting					
	Transfer clock [Hz]	00 (No noise rejection)	01 (Reject pulses shorter than 31/fc[s] as noise)	Reject pulses shorter than 63/fc[s] as noise)	11 (Reject pulses shorter than 127/fc[s] as noise)		
000	fc/13	0	0	0	<u> </u>		
110	fc/8	0	(f/\wedge)	- 6	-		
(When the transfer clock gen- erated by timer/counter inter- rupt is the same as the right side column)	fc/16	0		\diamond -	<u> </u>		
	fc/32	0		0	-		
The setting except the	above	0 <		$(\circ))$	0		

Table of Contents

Differences among Products	
TMP86FS49BUG	
1.1Features1.2Pin Assignment1.3Block Diagram1.4Pin Names and Functions	$\begin{array}{c} \hline \\ \hline $
2. Operational Description	
2.1 CPU Core Functions 2.1.1 Memory Address Map 2.1.2 Program Memory (Flash) 2.1.3 Data Memory (RAM)	9 9 9 9 9
2.2 System Clock Controller	10 12 13
2.2.3.2 Duál-Clock móde 2.2.3.3 STOR mode 2.2.4 Operating Mode Control 2.2.4.1 STOP mode 2.2.4.2 IDLE1/2 mode and SLEEP1/2 mode 2.2.4.3 IDLE0 and SLEEP0 modes (IDLE0, SLEER0)	
2.2.43 SLOW mode 2.3 Reset Circuit	
3. Interrupt Control Circuit	

3.1 Interrupt latches (IL23 to IL2)	35
3.2 Interrupt enable register (EIR)	36
3.2.1 Interrupt master enable flag (IMF)	
3.2.2 Individual interrupt enable flags (EF23 to EF4)	
Note 3:	38
3.3 Interrupt Sequence	39
3.3.1 Interrupt acceptance processing is packaged as follows	
3.3.2 Saving/restoring general-purpose registers	
3.3.2.1 Using PUSH and POP instructions	



The interrupt latches are located on address 002EH, 003CH and 003DH in SFR area. Each latch can be cleared to "0" individually by instruction. However, IL2 and IL3 should not be cleared to "0" by software. For clearing the interrupt latch, load instruction should be used and then IL2 and IL3 should be set to "1". If the read-modify-write instructions such as bit manipulation or operation instructions are used, interrupt request would be cleared inadequately if interrupt is requested while such instructions are executed.

Interrupt latches are not set to "1" by an instruction.

Since interrupt latches can be read, the status for interrupt requests can be monitored by software.

Note: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction) In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on

interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

Example 1 :Clears interrupt latches	
DI	;/MJF ~ 9
LDW	(ILL), 1110100000111111B ; \L12, IL10 to IL6 ← 0 <
EI	
Example 2 :Reads interrupt latchess	
LD	WA, (ILL) ; W ← ILH, A ← ILL
Example 3 :Tests interrupt latches	
TEST	(ILL). 7 ; if IL7 = 1 then jump
JR	F, SSET
3.2 Interrupt enable regis	ster (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the non-maskable interrupts (Software interrupt, undefined instruction interrupt, address trap interrupt and watchdog interrupt). Non-maskable interrupt is accepted regardless of the contents of the EIR.

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located on address 002CH, 003AH and 003BH in SFR area, and they can be read and written by an instructions (Including read-modify-write instructions such as bit manipulation or operation instructions).

3.2.1 Interrupt master enable flag (IMF)

The interrupt enable register (IMF) enables and disables the acceptance of the whole maskable interrupt. While IMF = "0", all maskable interrupts are not accepted regardless of the status on each individual interrupt enable flag (EF). By setting IMF to "1", the interrupt becomes acceptable if the individuals are enabled. When an interrupt is accepted, IMF is cleared to "0" after the latest status on IMF is stacked. Thus the maskable interrupts which follow are disabled. By executing return interrupt instruction [RETI/RETN], the stacked data, which was the status before interrupt acceptance, is loaded on IMF again.

The IMF is located on bit0 in EIRL (Address: 003AH in SFR), and can be read and written by an instruction. The IMF is normally set and cleared by [EI] and [DI] instruction respectively. During reset, the IMF is initialized to "0".

4.2 DBR

Address	Read	Write
0F80H	Res	erved
0F81H	Res	erved
0F82H	Res	erved
0F83H	Res	erved
0F84H	Res	erved
0F85H	Res	erved
0F86H	Res	erved
0F87H	Res	erved
0F88H	Reso	erved
0F89H	Res	bevre
0F8AH	Res	arved
0F8BH	Res	erved
0F8CH	Res	erved
0F8DH	Res	erved
0F8EH	Res	erved
0F8FH	Rese	erved
0F90H	SBISRA	SBICRA
0F91H	SBI	DBR
0F92H		I2CAR
0F93H	SBISRB	SBICRB
0F94H	Res	erved
0F95H	UART1SR	UART1CR1
0F96H		UART1CR2
0F97H) RD1BUF	TD1BUF
0F98H	UART2SR	UART2CR1
0F99Н		UART2CR2
0F9AH	RD2BUF	TD2BUF
ОЕЭВН	P60	CR1
OF9CH	P60	CR2
0F9DH	P70	CR1
ОГ9ЕН	P7(CR2
0F9FH	\sim $$	STOPCR
Address	Pool	\ \ /-:+~
Address	Read	write
	Res	
		onvod
	Res	51 VGU
Address	Read	Write
OFCOH	Resi	erved
0FDFH	Reso	erved
-		



9. 16-Bit Timer/Counter2 (TC2)

9.1 Configuration



10.2 TimerCounter Control

The TimerCounter 3 is controlled by the TimerCounter 3 control register (TC3CR) and two 8-bit timer registers (TTREG3, PWREG3).



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

- Note 2: Do not change the TC3M, TC3CK and TFF3 settings while the timer is running.
- Note 3: To stop the timer operation (TC3S= $1 \rightarrow 0$), do not change the TC3M, TC3CK and TFF3 settings. To start the timer operation (TC3S= $0 \rightarrow 1$), TC3M, TC3CK and TFF3 can be programmed.
- Note 4: To use the TimerCounter in the 16-bit mode, set the operating mode by programming TC4CR<TC4M>, where TC3M must be fixed to 011.
- Note 5: To use the TimerCounter in the 16-bit mode, select the source clock by programming TC3CK. Set the timer start control and timer F/F control by programming TC4CR<TC4S> and TC4CR<TFF4>, respectively.
- Note 6: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-1 and Table 10-2.

10.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4)

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 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 1/0 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 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 PWMj pin to the high level when the TimerCounter is stopped CLR (TCjCR).3: Stops the timer.

- CLR (TCjCR).7: Sets the 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 PWMj pin during the warm-up period time atter exiting the STOP mode.
- Note 4: j = 3, 4

Source Clock			Resolution		Repeated Cycle	
NORMAL1/2, DV7CK = 0	IDLE1/2 mode	SLOW1/2, SLEEP1/2 mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
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	$ \leq ($	8 μs	-	2.05 ms	-
fc/25) 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	_

Table 10-5 PWM Output Mode

10.3.8 16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4)

This mode is used to generate pulses with up to 16-bits of resolution. The timer counter 3 and 4 are cascadable to enter the 16-bit PPG mode.

The counter counts up using the internal clock or external clock. When a match between the up-counter and the timer register (PWREG3, PWREG4) value is detected, the logic level output from the timer F/F4 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F4 is switched to the opposite state again when a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected, and the counter is cleared. The INTTC4 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC3 pin. Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F4 by TC4CR<TFF4>, positive and negative pulses can be generated. Upon reset, the timer F/F4 is cleared to 0.

(The logic level output from the $\overline{PPG4}$ pin is the opposite to the timer F/F4.)

Set the lower byte and upper byte in this order to program the timer register. (TTREG3 \rightarrow TTREG4, PWREG3 \rightarrow PWREG4) (Programming only the upper or lower byte should not be attempted.)

For PPG output, set the output latch of the I/O port to 1.

Example :Generating a pulse with 1-ms high-level width and a period of 16.385 ms (fc = 16.0 MHz)

	Setting ports	
LDW	(PWREG3), 07D0H	: Sets the pulse width.
LDW	(TTREG3), 8002H	: Sets the cycle period.
LD	(TC3GR), 33H	: Sets the operating clock to fc/2 ³ , and 16-bit PPG mode (lower byte).
LD	(TC4CR), 057H	: Sets TFF4 to the initial value 0, and 16-bit PPG mode (upper byte).
	(TC4CR), 05FH	: Starts the timer.
$\langle \rangle$	9 6	$7 \wedge$

- Note 1: In the PPG mode, do not change the PWREG and TREG settings while the timer is running. Since PWREG and TTREG are not in the shift register configuration in the PPG mode, the new values programmed in PWREG and TTREG are in effect immediately after programming PWREG and TTREG. Therefore, if PWREG and TTREG are changed while the timer is running, an expected operation may not be obtained.
- Note 2: When the timer is stopped during PPG output, the PPG4 pin holds the output status when the timer is stopped. To change the output status, program TC4CR<TFF4> after the timer is stopped. Do not change TC4CR<TFF4> upon stopping of the timer.

Example: Fixing the PPG4 pin to the high level when the TimerCounter is stopped

- CLR (TC4CR).3: Stops the timer
 - CLR (TC4CR).7: Sets the PPG4 pin to the high level

Note 3: i = 3, 4

11.3.5 16-Bit Timer Mode (TC5 and 6)

In the timer mode, the up-counter counts up using the internal clock. The TimerCounter 5 and 6 are cascadable to form a 16-bit timer.

When a match between the up-counter and the timer register (TTREG5, TTREG6) value is detected after the timer is started by setting TC6CR<TC6S> to 1, an INTTC6 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter continues counting. Program the lower byte and upper byte in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the PDOj, PWMj, and PPGi pins may output a pulse.

Note 2: In the timer mode, do not change the TTREG setting while the timer is running. Since TTREG is not in the shift register configuration in the timer mode, the new value programmed in TTREG is in effect immediately after programming of TTREG. Therefore, if TTREG is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 5, 6

Table 11-6 Source Clock for 16-Bit Timer Mode

Source Clock			Resolution Maximum Time Set			ime Setting
NORMAL1/2,	IDLE1/2 mode	SLOW1/2,	($\langle \mathcal{O} \rangle$	\diamond	
DV7CK = 0	DV7CK = 1	SLEEP1/2 mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs/= 32.768 kHz
fc/2 ¹¹	fs/2 ³	fs/2 ³	128 µs	244.14 μs	8.39 s	16 s
fc/2 ⁷	fc/2 ⁷	-	8 µs		524.3 ms	-
fc/2 ⁵	fc/2 ⁵	_	2 µs	- (🤇	131.1 ms	_
fc/2 ³	fc/2 ³	- <	500 ns		32.8 ms	-

Example :Setting the timer mode with source clock $fe/2^7$ Hz, and generating an interrupt 300 ms later (fc = 16.0 MHz)



Figure 11-6 16-Bit Timer Mode Timing Chart (TC5 and TC6)



If received data is not read out from SIO1RDB receive error occurs immediately after shift operation is finished. Then INTSIO1 interrupt request is generated after SIO1SR<RXERR> is set to "1".

(3) Stopping the receive operation

There are two ways for stopping the receive operation.

- The way of clearing SIO1CR<SIOS>. When SIO1CR<SIOS> is cleared to "0", receive operation is stopped after all of the data is finished to receive. When receive operation is finished, SIO1SR<SIOF> is cleared to "0". In external clock operation, SIO1CR<SIOS> must be cleared to "0" before SIO1SR<SEF> is set to "1" by starting the next shift operation.
- The way of setting SIO1CR<SIOINH>. 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.





Figure 14-17 Hold Time of the End of Transmit/Receive Mode

(2) LSB receive mode

LSB receive mode is selected by setting SIO2CR<SIODIR> to "1", in which case the data is received sequentially beginning with the least significant bit (Bit0).

15.3.2.3 Transmit/receive mode

(1) MSB transmit/receive mode

MSB transmit/receive mode are selected by setting SIO2CR SIODIR> to "0" in which case the data is transferred sequentially beginning with the most significant bit (Bit7) and the data is received sequentially beginning with the most significant (Bit7).

(2) LSB transmit/receive mode

LSB transmit/receive mode are selected by setting SIO2CR<SIODIR to "1", in which case the data is transferred sequentially beginning with the least significant bit (Bit0) and the data is received sequentially beginning with the least significant (Bit0).

15.3.3 Transfer modes

Transmit, receive and transmit/receive mode are selected by using \$102CR<SIOM>.

15.3.3.1 Transmit mode

Transmit mode is selected by writing "00B" to SIO2CR<SIOM>.

(1) Starting the transmit operation

Transmit mode is selected by setting '00B' to SIO2CR<SIOM>. Serial clock is selected by using SIO2CR<SCK>. Transfer direction is selected by using SIO2CR<SIODIR>.

When a transmit data is written to the transmit buffer register (SIO2TDB), SIO2SR<TXF> is cleared to "0"

After SIO2CR<SIOS> is set to "1", SIO2SR<SIOF> is set synchronously to "1" the falling edge of SCK2 pin.

The data is transferred sequentially starting from SO2 pin with the direction of the bit specified by SIO2CR<SIODIR, synchronizing with the SCK2 pin's falling edge.

SIO2SR SEF is kept in high level, between the first clock falling edge of SCK2 pin and eighth clock falling edge.

SIO2SR $\langle TXF \rangle$ is set to "1" at the rising edge of pin after the data written to the SIO2TDB is transferred to shift register, then the INTSIO2 interrupt request is generated, synchronizing with the next falling edge on $\overline{SCK2}$ pin.

- Note 1: In internal clock operation, when SIO2CR<SIOS> is set to "1", transfer mode does not start without writing a transmit data to the transmit buffer register (SIO2TDB).
- Note 2: In internal clock operation, when the SIO2CR<SIOS> is set to "1", SIO2TDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from SCK2 pin.
- Note 3: In external clock operation, when the falling edge is input from SCK2 pin after SIO2CR<SIOS> is set to "1", SIO2TDB is transferred to shift register immediately.

If received data is not read out from SIO2RDB receive error occurs immediately after shift operation is finished. Then INTSIO2 interrupt request is generated after SIO2SR<RXERR> is set to "1".

(3) Stopping the receive operation

There are two ways for stopping the receive operation.

- The way of clearing SIO2CR<SIOS>. When SIO2CR<SIOS> is cleared to "0", receive operation is stopped after all of the data is finished to receive. When receive operation is finished, SIO2SR<SIOF> is cleared to "0". In external clock operation, SIO2CR<SIOS> must be cleared to "0" before SIO2SR<SEF> is set to "1" by starting the next shift operation.
- The way of setting SIO2CR<SIOINH>. Receive operation is stopped immediately after SIO2CR<SIOINH> is set to "1". In this case, SIO2CR<SIOS>, SIO2SR register, SIO2RDB register and SIO2TDB register are initialized.



16.6 Data Transfer of I²C Bus

16.6.1 Device initialization

For initialization of device, set the ACK in SBICRA to "1" and the BC to "000". Specify the data length to 8 bits to count clocks for an acknowledge signal. Set a transfer frequency to the SCK in SBICRA.

Next, set the slave address to the SA in I2CAR and clear the ALS to "0" to set an addressing format.

After confirming that the serial bus interface pin is high level, for specifying the default setting to a slave receiver mode, clear "0" to the MST, TRX and BB in SBICRB, set "1" to the PIN, "10" to the SBIM, and "00" to bits SWRST1 and SWRST0.

Note: The initialization of a serial bus interface circuit must be complete within the time from all devices which are connected to a bus have initialized to and device does not generate a start condition. If not, the data can not be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.

16.6.2 Start condition and slave address generation/

Confirm a bus free status (BB = 0).

Set the ACK to "1" and specify a slave address and a direction bit to be transmitted to the SBIDBR.

By writing "1" to the MST, TRX, BB and PIN, the start condition is generated on a bus and then, the slave address and the direction bit which are set to the SBIDBR are output. The time from generating the START condition until the falling SCL pin takes t_{IOW} :

An INTSBI interrupt request occurs at the 9th falling edge of a SCL clock cycle, and the PIN is cleared to "0". The SCL pin is pulled-down to the low level while the PIN is "0" When an interrupt request occurs, the TRX changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

- Note 1: Do not write a slave address to be output to the SBIDBR while data is transferred. If data is written to the SBIDBR, data to been outputting may be destroyed.
- Note 2: The bus free must be confirmed by software within 98.0 us (The shortest transmitting time according to the I²C bus standard) after setting of the slave address to be output. Only when the bus free is confirmed, set "1" to the MST, TRX, BB, and PIN to generate the start conditions. If the writing of slave address and setting of MST, TRX, BB and PIN doesn't finish within 98.0 us, the other masters may start the transferring and the slave address data written in SBIDBR may be broken.





16.6.3 1-word data transfer

Check the MST by the INTSBI interrupt process after an 1-word data transfer is completed, and determine whether the mode is a master or slave.

this case, FAH). If the 5th byte does not contain the operation command data, the device enters the halt condition after transmitting 3 bytes of operation command error code (63H).

- 4. The 7th through m'th bytes of the transmitted and received data contain the same data as in the flash memory writing mode.
- 5. The n'th byte contains the status to be transmitted to the external controller in the case of the successful security program.

20.9 Intel Hex Format (Binary)

- 1. After receiving the checksum of a data record, the device waits for the start mark (3AH ":") of the next data record. After receiving the checksum of a data record, the device ignores the data except 3AH transmitted by the external controller.
- 2. After transmitting the checksum of end record, the external controller must transmit nothing, and wait for the 2-byte receive data (upper and lower bytes of the checksum).
- 3. If a receiving error or Intel Hex format error occurs, the device enters the halt condition without returning an error code to the external controller. The Intel Hex format error occurs in the following case:

When the record type is not 00H, 01H, or 02H

When a checksum error occurs

When the data length of an extended record (record type = 02H) is not 02H

When the device receives the data record after receiving an extended record (record type = 02H) with extended address of 1000H or larger.

When the data length of the end record (record type = 01H) is not 00H

20.10Passwords

The consecutive eight or more-byte data in the flash memory area can be specified to the password. TMP86FS49BUG compares the data string specified to the password with the password string transmitted from the external controller. The area in which passwords can be specified is located at addresses 1000H to FF9FH. The area from FFA0H to FFFFH can not be specified as the passwords area.

If addresses from FFE0H through FFFFH are filled with "FFH", the passwords are not compared because the product is considered as a blank product. Even in this case, the password count storage addresses and password comparison start address must be specified. Table 20-16 shows the password setting in the blank product and non-blank product.

Table 20-16 Password Setting in the Blank Product and Non-Blank Product

Password	Blank Product (Note 1)	Non-Blank Product
PNSA (Password count storage address)	1000H SPNSA SFF9FH	$1000H \le PNSA \le FF9FH$
PCSA (Password comparison start address)	1000H≤ PCSA ≤ FF9FH	$1000H \le PCSA \le FFA0 - N$
N (Password count)	*	8 ≤ N
Password string setting	Not required (Note 5)	Required (Note 2)

Note 1: When addresses from FFE0H through FFFFH are filled with "FFH", the product is recognized as a blank product.

Note 2: The data including the same consecutive data (three or more bytes) can not be used as a password. (This causes a password error data. TMR86FS49BUG transmits no data and enters the halt condition.)

Note 6: In the flash memory erasing mode, the external controller must not transmit the password string for the blank product.

Note 3: *: Don't care.

Note 4: When the above condition is not met, a password error occurs. If a password error occurs, the device enters the halt condition without returning the error code.

Note 5: In the flash memory writing mode or RAM loader mode, the blank product receives the Intel Hex format data immediately after receiving PCSA without receiving password strings. In this case, the subsequent processing is performed correctly because the blank product ignores the data except the start mark (3AH ":") as the Intel Hex format data, even if the external controller transmits the dummy password string. However, if the dummy password string contains "3AH", it is detected as the start mark erroneously. The microcontroller enters the halt mode. If this causes the problem, do not transmit the dummy password strings.

22.5 AC Characteristics

(V_{SS} = 0 V, 4.5 V \leq V_{DD} $\leq~$ 5.5 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Тур.	Max	Unit
		NORMAL1, 2 modes	0.25	-	4	- μ s
Machina cycla timo	tov	IDLE0, 1, 2 modes	0.25			
	icy	SLOW1, 2 modes	117.6		133.3	
		SLEEP0, 1, 2 modes			155.5	
High-level clock pulse width	t _{WCH}	For external clock operation (XIN input)	(//	21 25		200
Low-level clock pulse width	t _{WCL}	fc = 16 MHz		31.23	_	115
High-level clock pulse width	t _{WSH}	For external clock operation (XTIN input)	$\left(\right) \right\rangle$	15.26		
Low-level clock pulse width	t _{WSL}	fs = 32.768 kHz		13.20		μs

 $(V_{SS} = 0 \text{ V}, 2.7 \text{ V} \le V_{DD} < 4.5 \text{ V}, \text{ Topr} = -40 \text{ to } 85^{\circ}\text{C})$

Paramete	Symbol	Condition	Min	Тур.	Max	Unit
Machine cycle time		NORMAL1, 2 modes	\diamond			
	t _{cy}	IDLE0, 1, 2 modes	0.5	- 4		μs
		SLOW1, 2 modes	1170	1170		
		SLEEP0, 1, 2 modes	117.0	$\mathcal{D}\mathcal{F}$	133.3	
High-level clock pulse width	t _{WCH}	For external clock operation (XIN input)	$(\overline{\Omega})$	62.5		20
Low-level clock pulse width	t _{WCL}	fc = 8 MHz	()	02.5	-	115
High-level clock pulse width	t _{WSH}	For external clock operation (XTIN input)		15.00		
Low-level clock pulse width	t _{WSL}	fs = 32.768 kHz))	15.20	_	μs

22.6 Flash Characteristics

22.6.1 Write/Retention Characteristics

(Vss	=	0	V)

Paramete	Condition	Min	Тур.	Max.	Unit
Number of guaranteed writes to flash memory	V _{SS} = 0 V, Topr = -10 to 40°C	-	_	100	Times
	4				