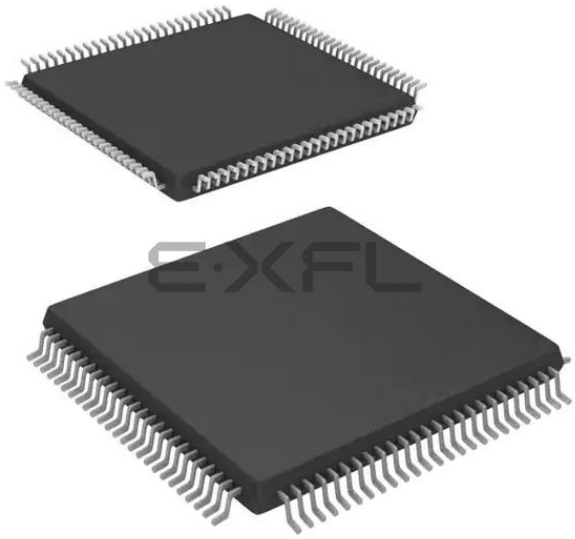

Welcome to E-XFL.COM

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	Obsolete
Core Processor	SH-2
Core Size	32-Bit Single-Core
Speed	20MHz
Connectivity	EBI/EMI, SCI
Peripherals	POR, PWM
Number of I/O	11
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/hd6417011vx20v

Interrupt Controller (INTC):

- Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$)
- Twenty-two internal interrupt sources
- Sixteen programmable priority levels

Bus State Controller (BSC):

- Supports external extended memory access
  - 16-bit external data bus
- Memory address space divided into four areas (four areas of SRAM space) with the following settable features:
  - Number of wait cycles (0 to 3 cycles)
  - Outputs chip-select signals for each area
- Wait cycles can be inserted using an external WAIT signal

Multifunction Timer Pulse Unit (MTU):

- Maximum 6 types of waveform output or maximum 6 types of pulse I/O processing possible based on 16-bit timer, 3 channels
- 8 dual-use output compare/input capture registers
- 8 independent comparators
- 6 types of counter input clock
- Input capture function
- Pulse output mode
  - One shot, toggle, PWM
- Multiple counter synchronization function

Compare Match Timer (CMT) (Two Channels):

- 16-bit free-running counter
- One compare register
- Generates an interrupt request upon compare match

8-Bit Timers (TIM1, TIM2) (Two Channels):

- 8-bit interval timer function
- Interrupt generated on counter overflow (TIM1)
- Interrupt generated on compare match (TIM2)

Table 2.11 Instruction Code Format

Item	Format	Explanation
Instruction	`OP.Sz SRC,DEST`	OP: Operation code Sz: Size (B: byte, W: word, or L: longword) SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*1
Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 . . . 1111: R15 iiii: Immediate data dddd: Displacement
Operation	→, ←	Direction of transfer
	(xx)	Memory operand
	M/Q/T	Flag bits in the SR
	&	Logical AND of each bit
	|	Logical OR of each bit
	^	Exclusive OR of each bit
	~	Logical NOT of each bit
	<<n	n-bit left shift
	>>n	n-bit right shift
Execution cycles	—	Value when no wait states are inserted*2
T bit	—	Value of T bit after instruction is executed. An em-dash (—) in the column means no change.

Notes: 1. Depending on the operand size, displacement is scaled ×1, ×2, or ×4. For details, see the *SH-1/SH-2/SH-DSP Programming Manual*.

2. Instruction execution cycles: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.

5.1.2 Exception Processing Operations

The exception processing sources are detected and begin processing according to the timing shown in table 5.2.

Table 5.2 Timing of Exception Source Detection and the Start of Exception Processing

Exception	Source	Timing of Source Detection and Start of Processing
Power-on reset		Starts when the $\overline{\text{RES}}$ pin changes from low to high.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except after a delayed branch instruction (delay slot).
	Illegal slot instructions	Starts from the decoding of undefined code placed in a delayed branch instruction (delay slot) or of instructions that rewrite the PC.

When exception processing starts, the CPU operates as follows:

1. Exception processing triggered by reset:

   The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception processing vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses). See section 5.1.3, Exception Processing Vector Table, for more information. 0 is then written to the vector base register (VBR) and 1111 is written to the interrupt mask bits (I3–I0) of the status register (SR). The program begins running from the PC address fetched from the exception processing vector table.
2. Exception processing triggered by address errors, interrupts and instructions:

   SR and PC are saved to the stack indicated by R15. For interrupt exception processing, the interrupt priority level is written to the SR's interrupt mask bits (I3–I0). For address error and instruction exception processing, the I3–I0 bits are not affected. The start address is then fetched from the exception processing vector table and the program begins running from that address.

5.1.3 Exception Processing Vector Table

Before exception processing begins running, the exception processing vector table must be set in memory. The exception processing vector table stores the start addresses of exception service routines. (The reset exception processing table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception processing, the start addresses of the exception service routines are fetched from the exception processing vector table, which indicated by this vector table address.

Table 5.3 shows the vector numbers and vector table address offsets. Table 5.4 shows how vector table addresses are calculated.

Table 5.3 Exception Processing Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset
Power-on reset	PC	0	H'00000000–H'00000003
	SP	1	H'00000004–H'00000007
(Reserved by system)		2	H'00000008–H'0000000F
		3	
General illegal instruction		4	H'00000010–H'00000013
(Reserved by system)		5	H'00000014–H'00000017
Slot illegal instruction		6	H'00000018–H'0000001B
(Reserved by system)		7	H'0000001C–H'0000001F
(Reserved by system)		8	H'00000020–H'00000023
CPU address error		9	H'00000024–H'00000027
(Reserved by system)		10	H'00000028–H'0000002B
Interrupts	NMI	11	H'0000002C–H'0000002F
(Reserved by system)		12	H'00000030–H'00000033
		:	:
		31	H'0000007C–H'0000007F
Trap instruction (user vector)		32	H'00000080–H'00000083
		:	:
		63	H'000000FC–H'000000FF

7.3.3 $\overline{CS}$ Assert Period Extension

Idle cycles can be inserted to prevent extension of the $\overline{RD}$ signal or $\overline{WRx}$ signal assert period beyond the length of the $\overline{CSn}$ signal assert period by setting the SW3–SW0 bits of BCR2. This allows for flexible interfaces with external circuitry. The timing is shown in figure 7.6. T_h and T_f cycles are added respectively before and after the ordinary cycle. Only $\overline{CSn}$ is asserted in these cycles; $\overline{RD}$ and $\overline{WRx}$ signals are not. Further, data is extended up to the T_f cycle, which is effective for gate arrays and the like, which have slower write operations.

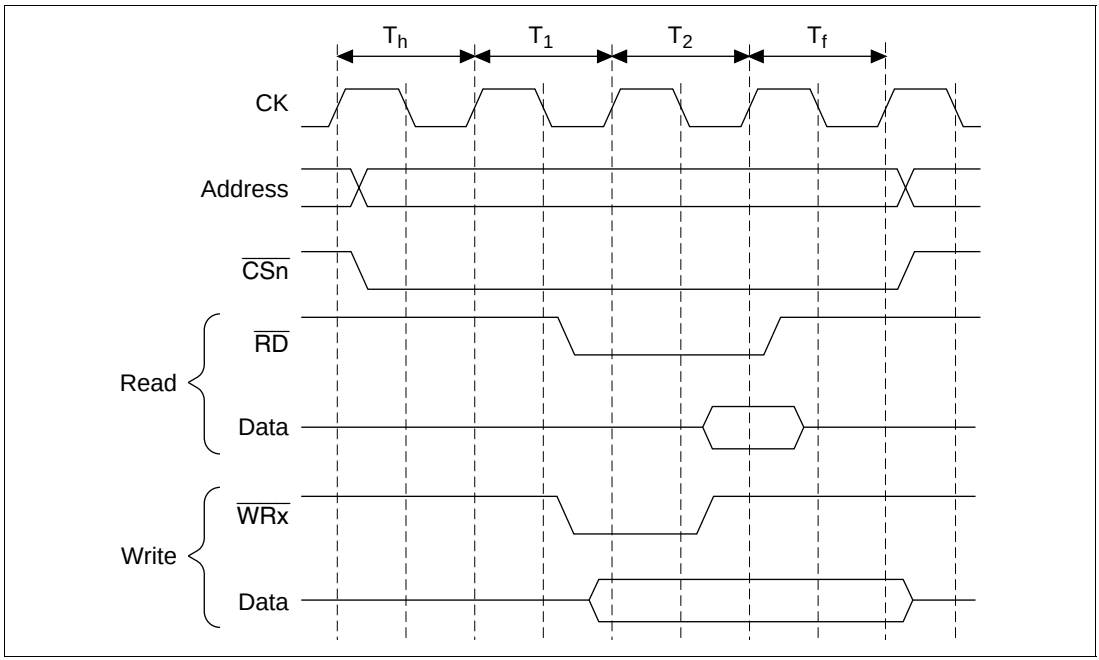

Figure 7.6 $\overline{CS}$ Assert Period Extension Function

7.5 Memory Connection Examples

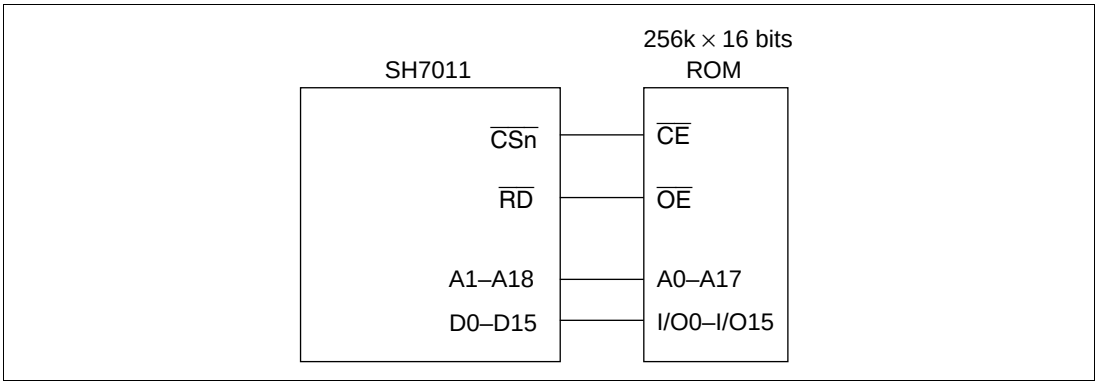

Figure 7.9 16-Bit Data Bus Width ROM Connection

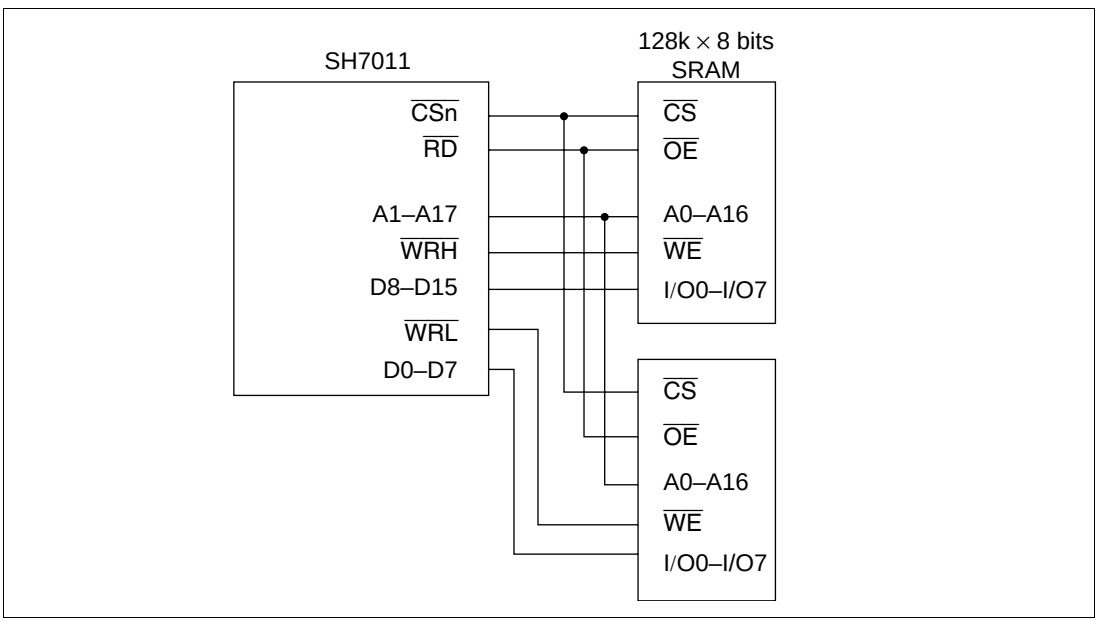

Figure 7.10 16-Bit Data Bus Width SRAM Connection

Bit n: CSTn	Description
0	TCNTn count is halted (initial value)
1	TCNTn counts

Note: n = 2 to 0.

If 0 is written to a CST bit during operation with the TIOC pin in the output state, the counter stops, but the TIOC pin output compare output level is maintained. If a write is performed on the TIOR register while a CST bit is 0, the pin output level is updated to the set initial output value.

8.2.9 Timer Synchro Register (TSYR)

The timer synchro register (TSYR) is an 8-bit read/write register that selects independent or synchronous TCNT counter operation for channels 0–2. Channels for which 1 is set in the corresponding bit will be synchronized. TSYR is initialized to H'00 upon power-on reset.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	SYNC2	SYNC1	SYNC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

- Bits 7–3—Reserved. These bits always read 0. The write value should always be 0.
- Bits 2–0—Timer Synchronization 2–0 (SYNC2–SYNC0): Selects operation independent of, or synchronized to, other channels. Synchronous operation allows synchronous clears due to multiple TCNT synchronous presets and other channel counter clears. A minimum of two channels must have SYNC bits set to 1 for synchronous operation. For synchronization clearing, it is necessary to set the TCNT counter clear sources (the CCLR2–CCLR0 bits of the TCR register), in addition to the SYNC bit. The counter start to channel and bit-to-channel correspondence are indicated in the tables below.

  SYNC2: Channel 2 (TCNT2)

  SYNC1: Channel 1 (TCNT1)

  SYNC0: Channel 0 (TCNT0)

The procedure for selecting the input capture operation (figure 8.12) is:

1. Set the TIOR to select the input capture function of the TGR, then select the input capture source, and rising edge, falling edge, or both edges as the input edge.
2. Set the CST bit in the TSTR to 1 to start the TCNT counting.

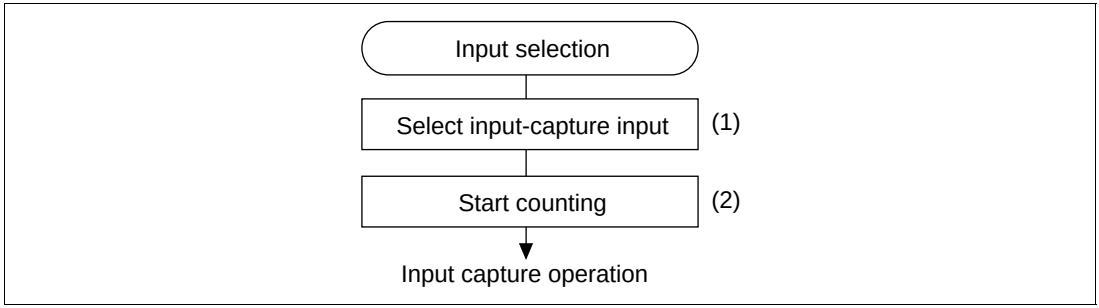

Figure 8.12 Procedure for Selecting Input Capture Operation

Input Capture Operation: Figure 8.13 shows input capture. The falling edge of TIOCB and both edges of TIOCA are selected as input capture input edges. In the example, TCNT is set to clear at the input capture of the TGRB register.

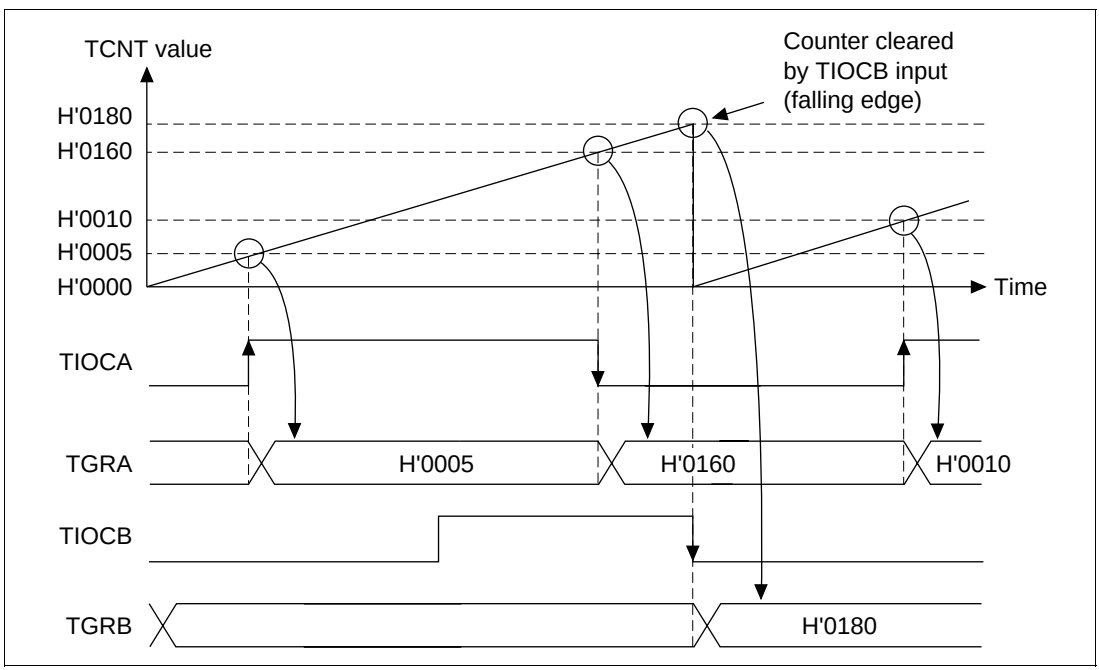

Figure 8.13 Input Capture Operation

(1) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode: Figure 8.49 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

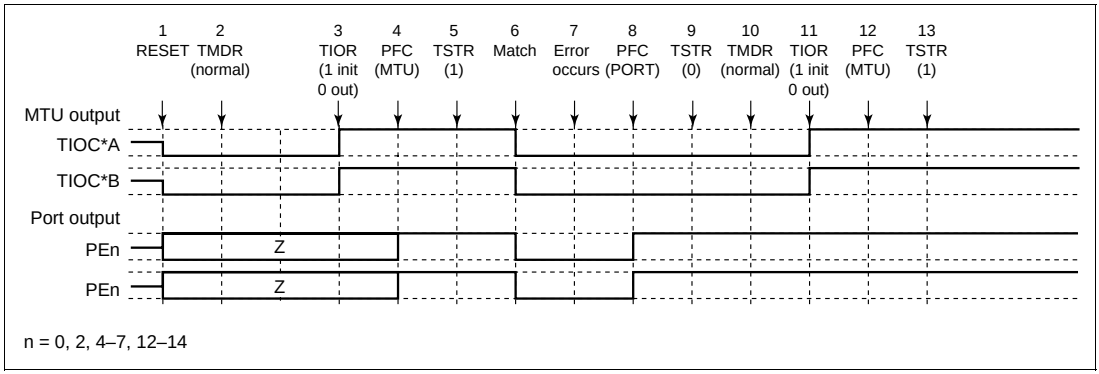

Figure 8.49 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set MTU output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Not necessary when restarting in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

(7) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode: Figure 8.55 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

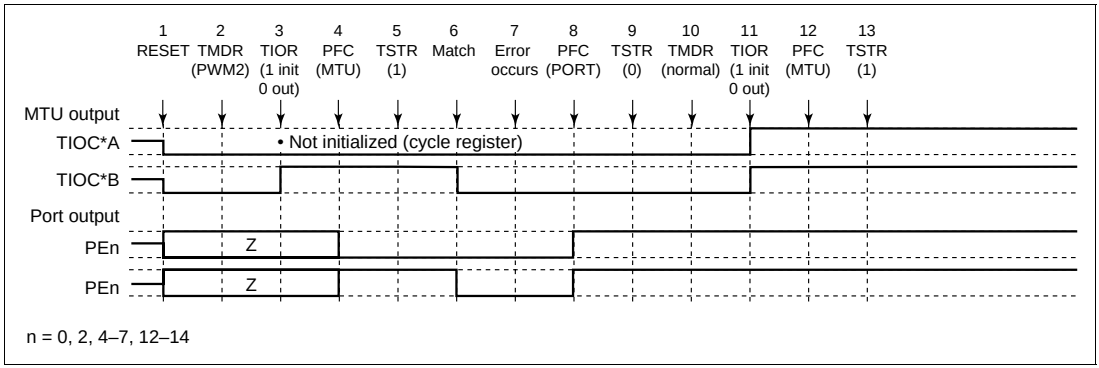

Figure 8.55 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, TIOC*A is the cycle register.)
4. Set MTU output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

9.1.2 Block Diagram

Figure 9.1 shows a block diagram of 8-bit timer 1 (TIM1).

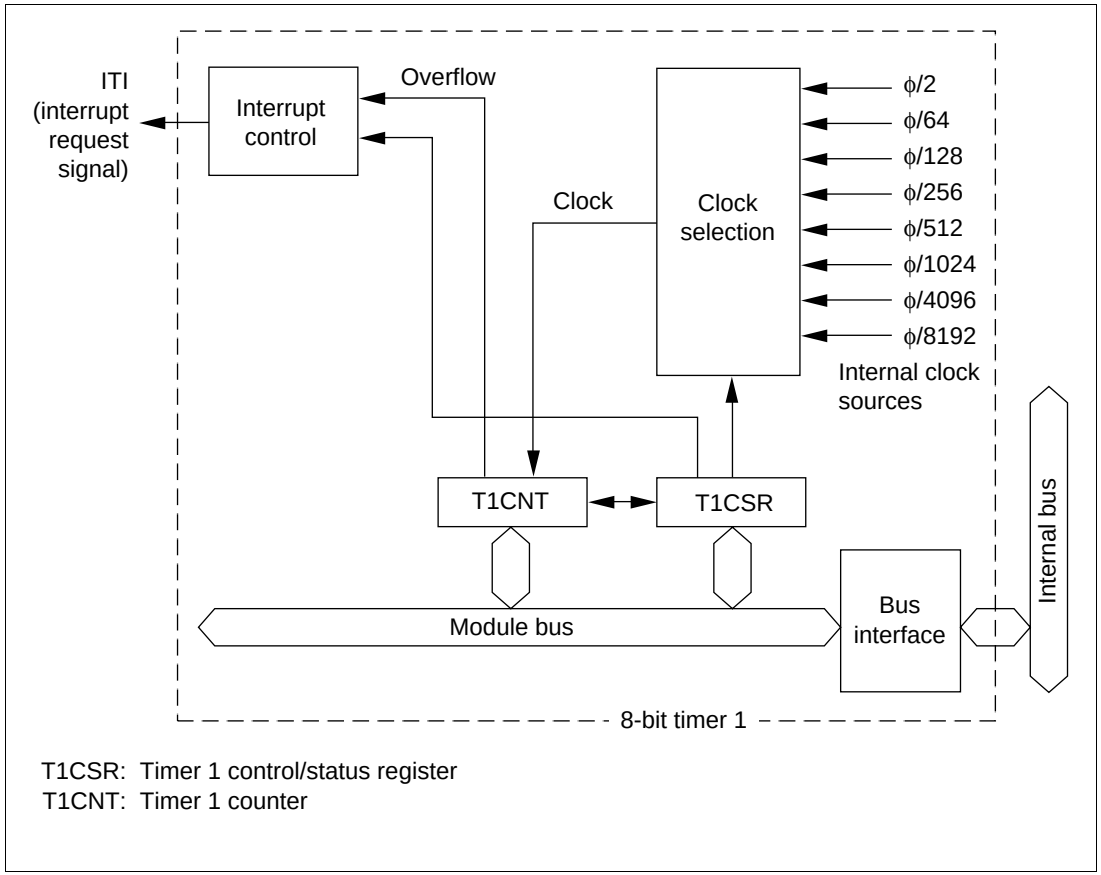

Figure 9.1 Block Diagram of 8-Bit Timer 1

10.4.3 Timing of Compare Match Flag Clearing

The CMF bit in the T2CSR register is cleared by reading the bit when it is set to 1, then writing 0 in it. Figure 10.5 shows the timing of CMF bit clearing by the CPU.

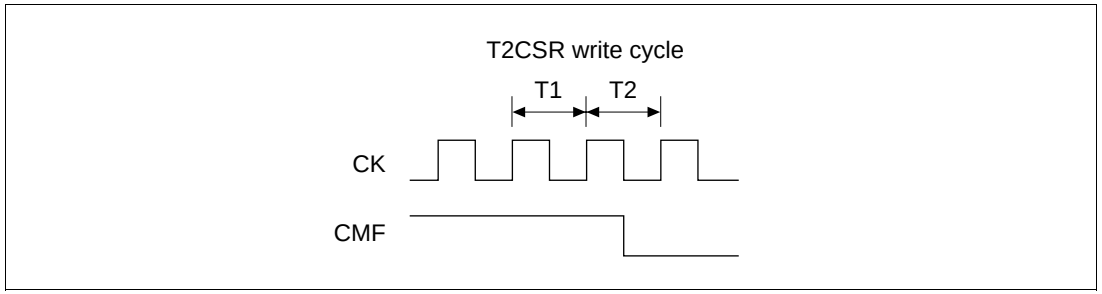

Figure 10.5 Timing of CMF Clearing by CPU

11.3.2 CMCNT Count Timing

One of four clocks (ϕ/8, ϕ/32, ϕ/128, ϕ/512) obtained by dividing the system clock (CK) can be selected by the CKS1, CKS0 bits of the CMCSR. Figure 11.3 shows the timing.

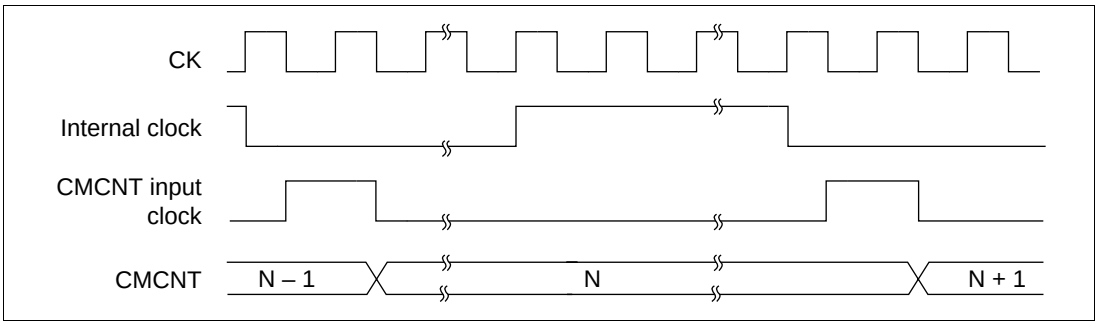

Figure 11.3 Count Timing

11.4 Interrupts

11.4.1 Interrupt Sources

The CMT has a compare match interrupt for each channel, with independent vector addresses allocated to each of them. The corresponding interrupt request is output when the interrupt request flag CMF is set to 1 and the interrupt enable bit CMIE has also been set to 1.

When activating CPU interrupts by interrupt request, the priority between the channels can be changed by using the interrupt controller settings. See section 6, Interrupt Controller, for details.

11.4.2 Compare Match Flag Set Timing

The CMF bit of the CMCSR register is set to 1 by the compare match signal generated when the CMCOR register and the CMCNT counter match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT counter matching count value is updated). Consequently, after the CMCOR register and the CMCNT counter match, a compare match signal will not be generated until a CMCNT counter input clock occurs. Figure 11.4 shows the CMF bit set timing.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	ϕ (initial value)
	1	ϕ/4
1	0	ϕ/16
	1	ϕ/64

12.2.6 Serial Control Register (SCR)

The serial control register (SCR) operates the SCI transmitter/receiver, enables/disables interrupt requests. The CPU can always read and write the SCR. The SCR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

- Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TxI) requested when the transmit data register empty bit (TDRE) in the serial status register (SSR) is set to 1 by transfer of serial transmit data from the TDR to the TSR.

Bit 7: TIE	Description
0	Transmit-data-empty interrupt request (TxI) is disabled (initial value). The TxI interrupt request can be cleared by reading TDRE after it has been set to 1, then clearing TDRE to 0, or by clearing TIE to 0.
1	Transmit-data-empty interrupt request (TxI) is enabled

- Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RxI) requested when the receive data register full bit (RDRF) in the serial status register (SSR) is set to 1 by transfer of serial receive data from the RSR to the RDR. It also enables or disables receive-error interrupt (ERI) requests.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RxI) and receive-error interrupt (ERI) requests are disabled (initial value). RxI and ERI interrupt requests can be cleared by reading the RDRF flag or error flag (FER, PER, or ORER) after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0.
1	Receive-data-full interrupt (RxI) and receive-error interrupt (ERI) requests are enabled.

12.3.2 Operation in Asynchronous Mode

In the asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 12.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the marking (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in the asynchronous mode, the SCI synchronizes on the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

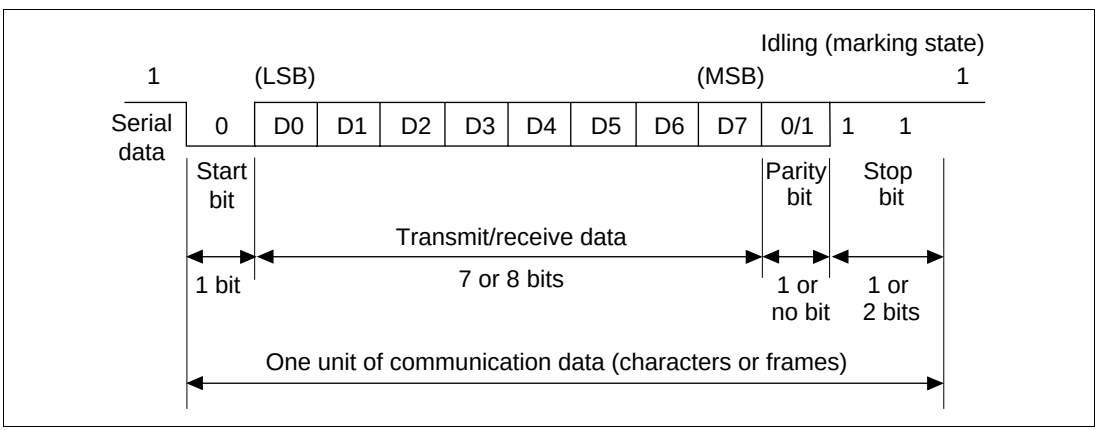

Figure 12.2 Data Format in Asynchronous Communication (Example: 8-bit Data with Parity and Two Stop Bits)

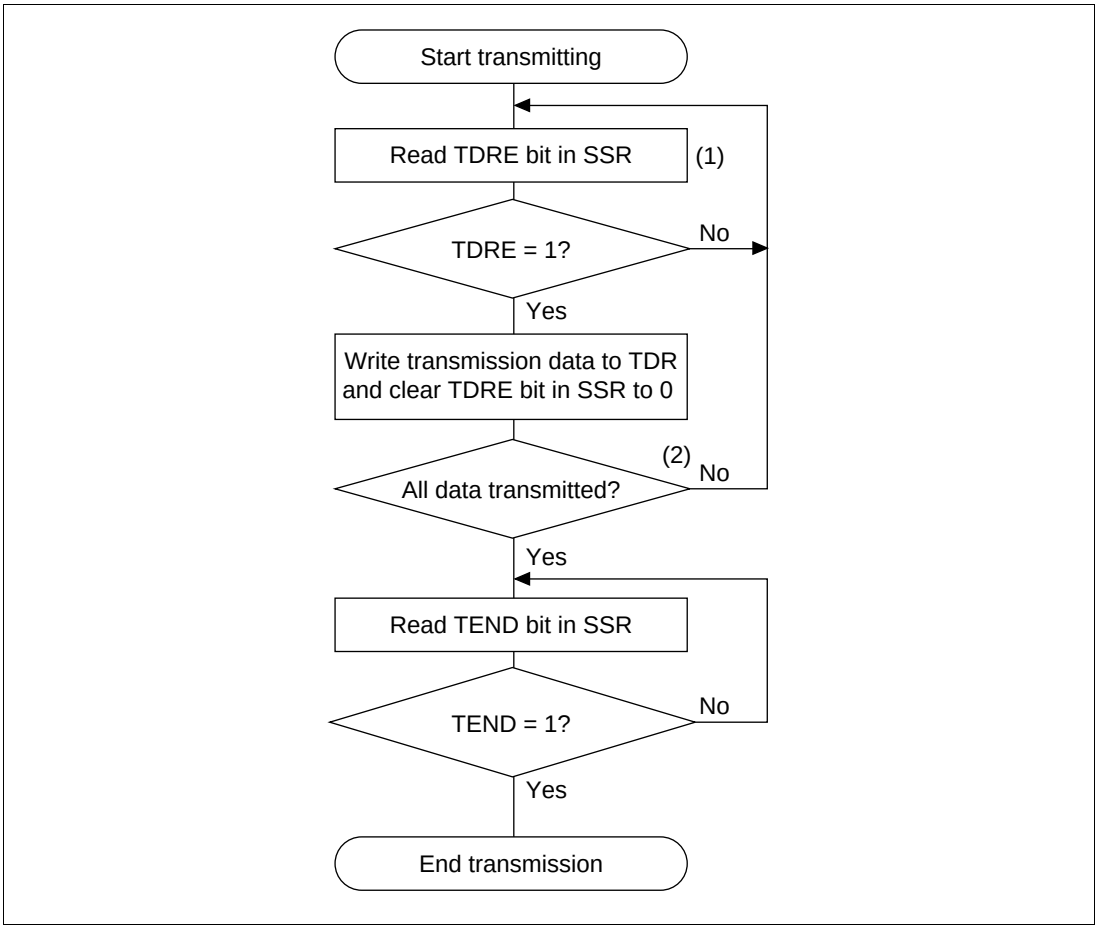

Figure 12.4 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SSR. When TDRE is cleared to 0, the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from the TDR into the transmit shift register (TSR).
2. After loading the data from the TDR into the TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in the SCR, the SCI requests a transmit-data-empty interrupt (TxI) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: one 0 bit is output.
   b. Transmit data: seven or eight bits of data are output, LSB first.
   c. Parity bit or multiprocessor bit: one parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.

13.1.2 Block Diagram

Figure 13.1 is the block diagram of the A/D converter.

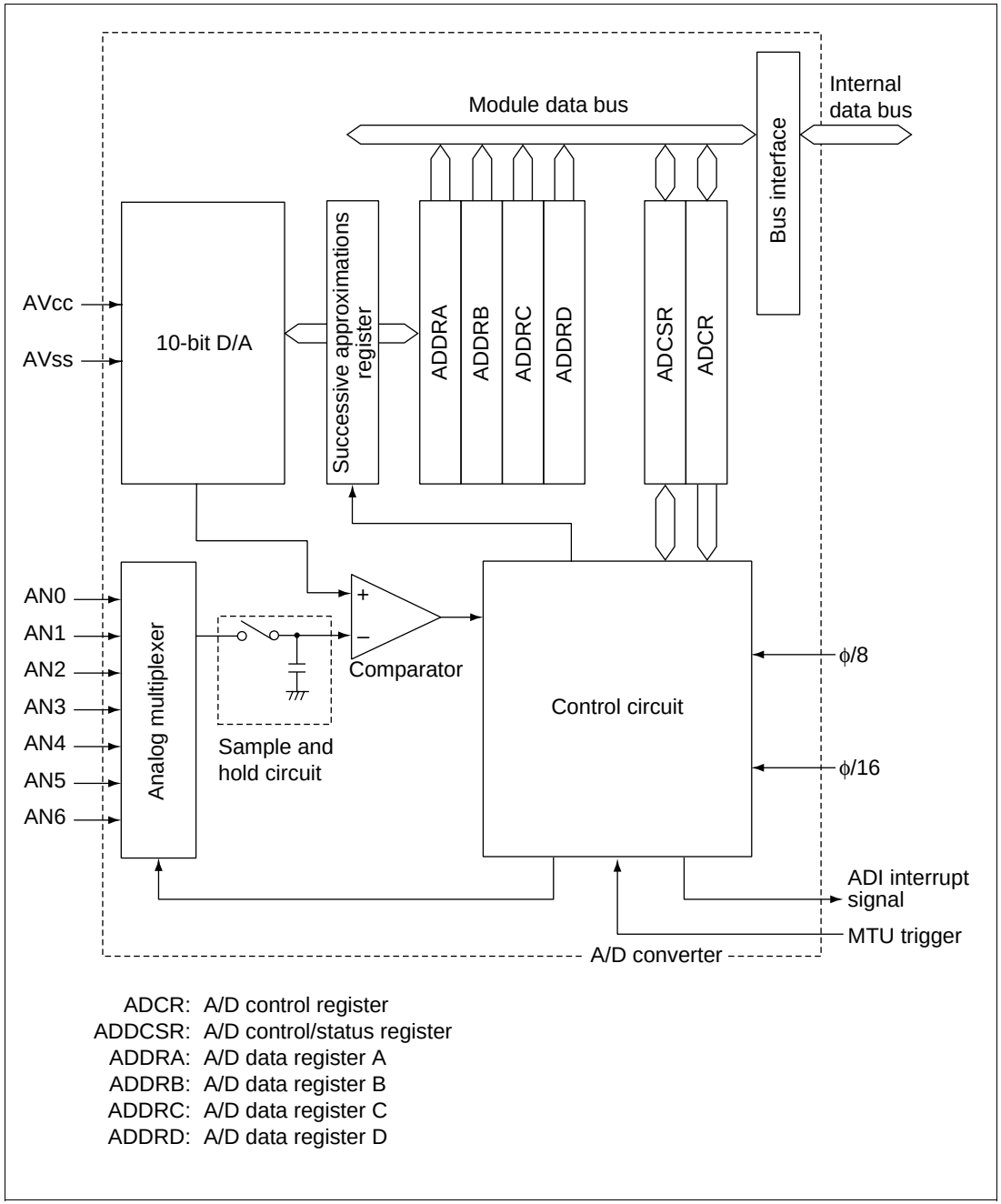

Figure 13.1 A/D Converter Block Diagram

Table 15.5 Correspondence between Port E Data Register (PEDR) Bits and Port E Pins

PEDR Bit	Port E Pin
PE14DR	PE14
PE13DR	PE13
PE12DR	PE12
PE7DR	PE7/TIOC2B
PE6DR	PE6/TIOC2A
PE5DR	PE5/TIOC1B
PE4DR	PE4/TIOC1A
PE2DR	PE2/TIOC0C
PE0DR	PE0/TIOC0A

17.4 A/D Converter Characteristics

Table 17.11 A/D Converter Characteristics (Conditions: V_{CC} = 3.0 to 3.6V, AV_{CC} = 3.0 to 5.5V, AV_{CC} V $_{CC}$, V_{SS} = AV_{SS} = 0V, T_a = Ð20 to +75°C, CKS = 0)

	20 MHz			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	bit
Conversion time	—	—	13.4	µs
Analog input capacity	—	—	20	pF
Permission signal source impedance	—	—	1	kΩ
Non-linearity error*	—	—	± 3	LSB
Offset error*	—	—	± 3	LSB
Full scale error*	—	—	± 3	LSB
Quantize error*	—	—	± 0.5	LSB
Absolute error	—	—	± 4	LSB

Note: * Reference values

Table 17.12 A/D Converter Characteristics (Conditions: V_{CC} = 3.0 to 3.6V, AV_{CC} = 3.0 to 5.5V, AV_{CC} V $_{CC}$, V_{SS} = AV_{SS} = 0V, T_a = Ð20 to +75°C, CKS = 1)

	20 MHz			
Item	**min**	**typ**	**max**	**Unit**
Resolution	10	10	10	bit
Conversion time	—	—	6.7	µs
Analog input capacity	—	—	20	pF
Permission signal source impedance	—	—	1	kΩ
Non-linearity error*	—	—	±5	LSB
Offset error*	—	—	±5	LSB
Full scale error*	—	—	±5	LSB
Quantize error*	—	—	±0.5	LSB
Absolute error	—	—	±6	LSB

Note: * Reference values

Appendix C Package Dimensions

Package dimensions of the SH7011 (TFP-100B) are shown in figure C.1.

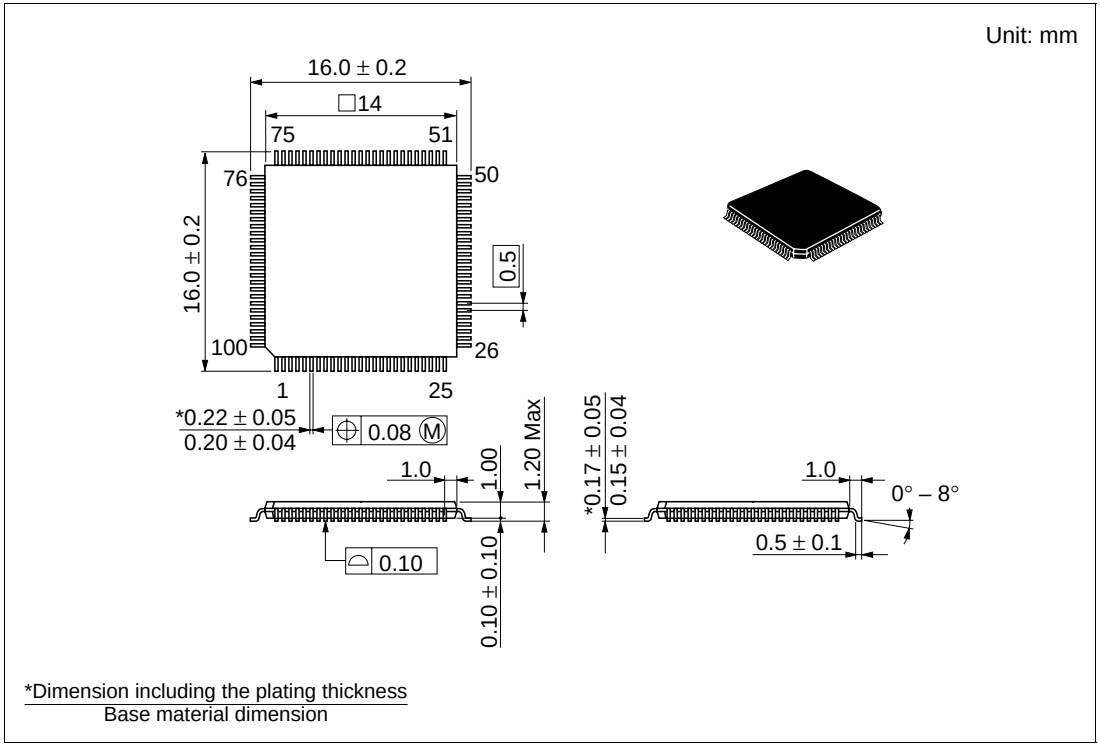

Figure C.1 Package Dimensions (TFP-100B)