TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/H Series

TMP95FY64

TOSHIBA CORPORATION

Preface

Thank you very much for making use of Toshiba microcomputer LSIs.
Before use this LSI, refer the section, "Points of Note and Restrictions".
Especially, take care below cautions.

CAUTION

How to release the HALT mode

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

CMOS 16-Bit Microcontroller

TMP95FY64F

1. Outline and Features

The TMP95FY64F is a high-speed 16-bit microcontroller developed for use in controlling various types of middle to large-sized equipment. With 256 Kbytes of flash memory included, it allows your programs to be erased and rewritten on-board.

This device comes in a 100-pin flat package. Its features are outlined below.

(1) High-speed 16-bit CPU (900/H CPU)

- Upward compatible in instruction mnemonics with the TLCS-90 / 900
- 16 Mbytes of linear addressing space
- General-purpose registers and register bank switching
- 16-bit divide / multiply instructions and bit transfer / bitwise operation instructions
- Micro DMA: 4 channels (2 bytes in 640 ns when using a 25 MHz oscillator)

(2) Minimum instruction execution time: 160 ns (when using a 25 MHz oscillator)

(3) Internal RAM: 8 Kbytes

Internal ROM: 256-Kbyte flash memory

2-Kbyte mask ROM (used for booting)

(4) External memory extension

- Expandable up to 16 Mbytes (shared between programs and data)
- External data bus width select pin (AM8/$\overline{16}$)
- Supports coexisting 8/16-bit external data buses .... Dynamic bus sizing

(5) 8-bit timer: 8 channels

- Event count function available (2 channels)

000707EBP1

- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance / Handling Precautions.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc..
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

(6) 16-bit timer / event counter: 2 channels

(7) General-purpose serial interface: 3 channels

(8) 10-bit A / D converter: 8 channels

(9) 8-bit D / A converter: 2 channels

(10) Watchdog timer

(11) Chip select / wait controller: 4 blocks

(12) Interrupt function: 45 lines of interrupt sources

- 9 CPU interrupts ...... Software interrupt instructions and undefined instruction execution
- 26 internal interrupts ┐ Seven priority levels can be set to resolve interrupt priority.
- 10 external interrupts ┘

(13) Input / output port: 81 pins

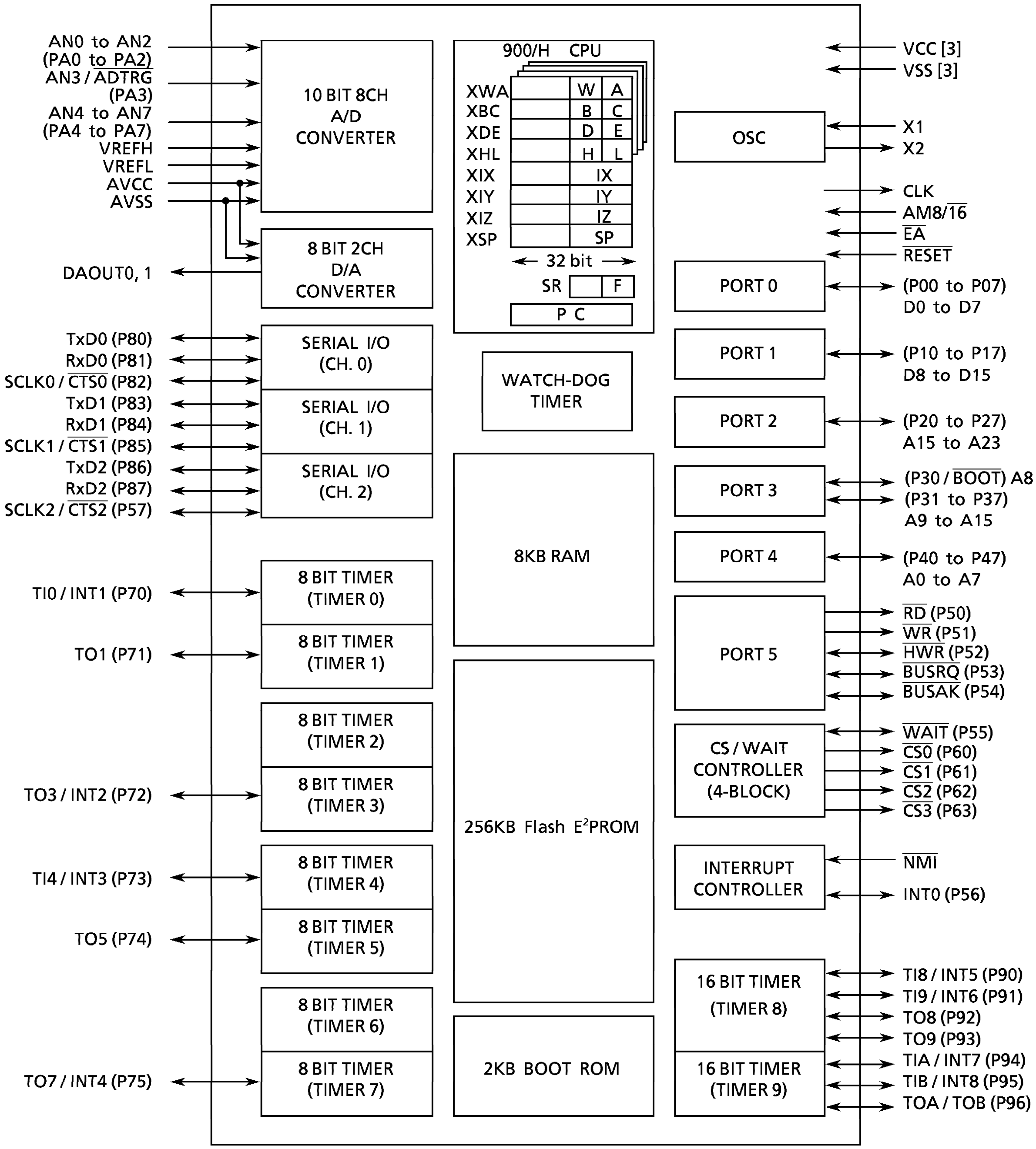
(14) Standby function

- Four halt modes (RUN, IDLE2, IDLE1, STOP)

(15) Operating voltage

- V_{CC} = 4.5 to 5.5 V

(16) Package: P-QFP100-1414-0.50E



Note: After a reset, the shared pins have their functions enclosed in ( ) selected.

Figure 1. Block Diagram of the TMP95FY64F

2. Pin Assignment and Functions

This section shows a pin assignment of the TMP95FY64F, as well as the names and the outline functions of its input / output pins.

2.1 Pin Assignment

The pin assignment of the TMP95FY64F is shown in Figure 2.1.

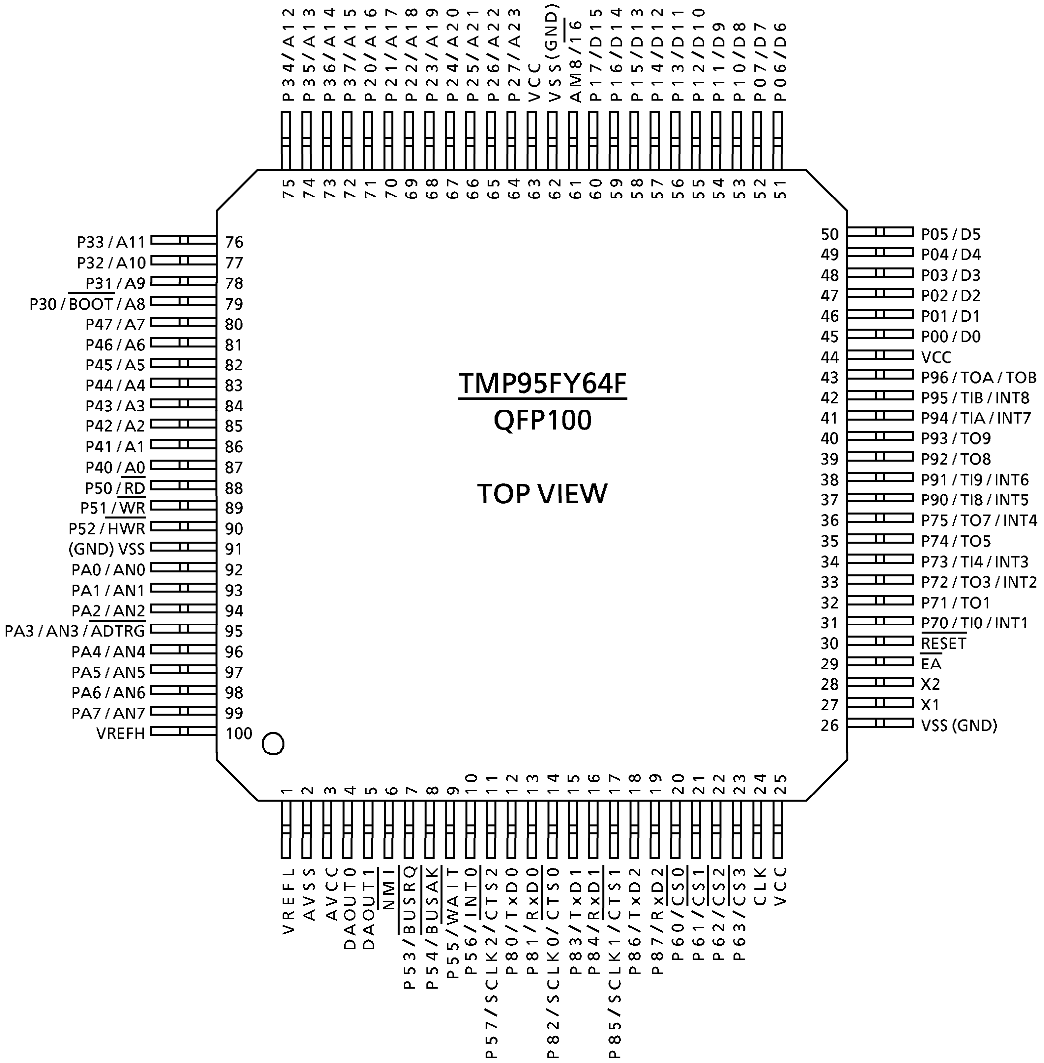


Figure 2.1 Pin Assignment of the TMP95FY64F (100-pin QFP)

2.2 Pin Names and Functions

The names and the outline functions of input/output pins are listed in Table 2.2.

Table 2.2 Pin Names and Functions (1/4)

Pin Name	No. of Pins	Type	Function
P00 to P07 /D0 to D7	8	Input/output	Port 0: Input/output port which can be set for input or output bit by bit.
		Input/output	Data: Data bus 0 to 7.
P10 to P17 /D8 to D15	8	Input/output	Port 1: Input/output port which can be set for input or output bit by bit.
		Input/output	Data: Data bus 8 to 15.
P20 to P27 /A16 to A23	8	Input/output	Port 2: Input/output port which can be set for input or output bit by bit.
		Output	Address: Address bus 16 to 23.
P30 /A8 /$\overline{\text{BOOT}}$	1	Input/output	Port 30: Input/output port. These pins should be pulled high with an external resistor.
		Output	Address: Address bus 8.
		Input	This pin sets single boot mode.
P31 to P37 /A9 to A15	7	Input/output	Ports 31 to 37: Input/output ports which can be set for input or output bit by bit.
		Output	Address: Address bus 9 to 15.
P40 to P47 /A0 to A7	8	Input/output	Port 4: Input/output port which can be set for input or output bit by bit.
		Output	Address: Address bus 0 to 7.
P50 /$\overline{\text{RD}}$	1	Output	Port 50: Output-only port.
		Output	Read: Strobe signal to read external memory. (The strobe signal can be output at any read timing by setting P5 <P50> = 0 and P5FC <P50F> = 1.)
P51 /$\overline{\text{WR}}$	1	Output	Port 51: Output-only port.
		Output	Write: Strobe signal to write data on pins D0 to 7.
P52 /$\overline{\text{HWR}}$	1	Input/output	Port 52: Input/output port (pull-up resistor included).
		Output	High-order write: Strobe signal to write on pins D8 to 15.
P53 /$\overline{\text{BUSRQ}}$	1	Input/output	Port 53: Input/output port (pull-up resistor included).
		Input	Bus request: Input pin to request that external bus be freed.
P54 /$\overline{\text{BUSAK}}$	1	Input/output	Port 54: Input/output port (pull-up resistor included).
		Output	Bus acknowledge: Output pin to notify that the CPU has freed external bus as requested by $\overline{\text{BUSRQ}}$.
P55 /$\overline{\text{WAIT}}$	1	Input/output	Port 55: Input/output port (pull-up resistor included).
		Input	Wait: Bus wait request to the CPU. (Effective in 1WAIT + N mode or 0 + N WAIT mode. This is set by the chip select/wait control register.)

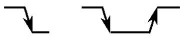
Table 2.2 Pin Names and Functions (2/4)

Pin Name	No. of Pins	Type	Function
P56	1	Input/output	Port 56: Input/output port (pull-up resistor included).
/INT0		Input	Interrupt request pin 0: Programmable (level or rising edge) interrupt request pin.
P57	1	Input/output	Port 57: Input/output port (pull-up resistor included).
/SCLK2		Input/output	Serial clock input/output 2
/$\overline{\text{CTS2}}$		Input	Serial data transmit ready 2 (Clear to Send)
P60	1	Output	Port 60: Output-only port.
/$\overline{\text{CS0}}$		Output	Chip select 0: Outputs a 0 when the address is within a specified address range.
P61	1	Output	Port 61: Output-only port.
/$\overline{\text{CS1}}$		Output	Chip select 1: Outputs a 0 when the address is within a specified address range.
P62	1	Output	Port 62: Output-only port.
/$\overline{\text{CS2}}$		Output	Chip select 2: Outputs a 0 when the address is within a specified address range.
P63	1	Output	Port 63: Output-only port.
/$\overline{\text{CS3}}$		Output	Chip select 3: Outputs a 0 when the address is within a specified address range.
P70	1	Input/output	Port 70: Input/output port.
/TI0		Input	Timer input 0: Input for timer 0.
/INT1		Input	Interrupt request pin 1: Rising edge interrupt request pin.
P71	1	Input/output	Port 71: Input/output port.
/TO1		Output	Timer output 1: Output for timer 0 or timer 1.
P72	1	Input/output	Port 72: Input/output port.
/TO3		Output	Timer output 3: Output for timer 2 or timer 3.
/INT2		Input	Interrupt request pin 2: Rising edge interrupt request pin.
P73	1	Input/output	Port 73: Input/output port.
/TI4		Input	Timer input 4: Input for timer 4.
/INT3		Input	Interrupt request pin 3: Rising edge interrupt request pin.
P74	1	Input/output	Port 74: Input/output port.
/TO5		Output	Timer output 5: Output for timer 4 or timer 5.
P75	1	Input/output	Port 75: Input/output port.
/TO7		Output	Timer output 7: Output for timer 6 or timer 7.
/INT4		Input	Interrupt request pin 4: Rising edge interrupt request pin.
P80	1	Input/output	Port 80: Input/output port (pull-up resistor included).
/TxD0		Output	Serial transmit data 0
P81	1	Input/output	Port 81: Input/output port (pull-up resistor included).
/RxD0		Input	Serial transmit data 0

Table 2.2 Pin Names and Functions (3/4)

Pin Name	No. of Pins	Type	Function
P82	1	Input/output	Port 82: Input/output port (pull-up resistor included).
/SCLK0		Input/output	Serial clock input/output 0
/$\overline{\text{CTS0}}$		Input	Serial data transmit ready 0 (Clear to Send).
P83	1	Input/output	Port 83: Input/output port (pull-up resistor included).
/TxD1		Output	Serial transmit data 1
P84	1	Input/output	Port 84: Input/output port (pull-up resistor included).
/RxD1		Input	Serial receive data 1
P85	1	Input/output	Port 85: Input/output port (pull-up resistor included).
/SCLK1		Input/output	Serial clock input/output 1
/$\overline{\text{CTS1}}$		Input	Serial data transmit ready 1 (Clear to Send)
P86	1	Input/output	Port 86: Input/output port (pull-up resistor included).
/TxD2		Output	Serial transmit data 2
P87	1	Input/output	Port 87: Input/output port (pull-up resistor included).
/RxD2		Input	Serial receive data 2
P90	1	Input/output	Port 90: Input/output port.
/TI8		Input	Timer input 8: Input pin for timer 8
/INT5		Input	Interrupt request pin 5: Programmable (rising or falling edge) interrupt request pin. ↑ ↓
P91	1	Input/output	Port 91: Input/output port.
/TI9		Input	Timer input 9: Input pin for timer 8
/INT6		Input	Interrupt request pin 6: Rising edge interrupt request pin. ↑
P92	1	Input/output	Port 92: Input/output port.
/TO8		Output	Timer output 8: Output pin for timer 8
P93	1	Input/output	Port 93: Input/output port.
/TO9		Output	Timer output 9: Output pin for timer 8
P94	1	Input/output	Port 94: Input/output port.
/TIA		Input	Timer input A: Input pin for timer 9
/INT7		Input	Interrupt request pin 7: Programmable (rising or falling edge) interrupt request pin. ↑ ↓
P95	1	Input/output	Port 95: Input/output port.
/TIB		Input	Timer input B: Input pin for timer 9
/INT8		Input	Interrupt request pin 8: Rising edge interrupt request pin. ↑
P96	1	Input/output	Port 96: Input/output port.
/TOA		Output	Timer output A: Output pin for timer 9
/TOB		Output	Timer output B: Output pin for timer 9
PA0 to PA2	3	Input	Port A0 to A2: Input-only port.
/AN0 to AN2		Input	Analog input 2 to 0: Input pin for the A/D converter.

Table 2.2 Pin Names and Functions (4/4)

Pin Name	No. of Pins	Type	Function
PA3	1	Input	Port A3: Input-only port.
/AN3		Input	Analog input 3: Input pin for the A/D converter.
/$\overline{\text{ADTRG}}$		Input	External start trigger
PA4 to PA7	4	Input	Port A4 to A7: Input-only port.
/AN4 to AN7		Input	Analog input 4 to 7: Input pin for the A/D converter.
DAOUT0	1	Output	D/A output 0: Output pin of D/A converter 0.
DAOUT1	1	Output	D/A output 1: Output pin of D/A converter 1.
$\overline{\text{NMI}}$	1	Input	Nonmaskable interrupt request pin: Programmable (rising edge or rising/falling edges) interrupt request pin. 
CLK	1	Output	Clock output: Clock derived from an external clock by dividing it by 4. This output is pulled high during a reset.
$\overline{\text{EA}}$	1	Input	External access: Connect this pin to Vcc.
AM8/$\overline{16}$	1	Input	Address mode: External data bus width select pin. Connect this pin to Vcc. The external data bus width to be accessed can be set by the chip select/wait control register.
$\overline{\text{RESET}}$	1	Input	Reset: Initializes the TMP95FY64. (Pull-up resistor included.)
VREFH	1	Input	Reference voltage input pin (H) for the A/D converter.
VREFL	1	Input	Reference voltage input pin (L) for the A/D converter.
AVCC	1		Power supply pin as well as the reference voltage input pin for the A/D converter. (Be sure to connect this pin to the power supply rail.)
AVSS	1		GND pin as well as the reference voltage input pin for the A/D converter. (Be sure to connect this pin to GND.)
X1/X2	2	Input/Output	Oscillator connecting pin.
VCC	3		Power supply pin: Connect all VCC pins to the power supply rail.
VSS	3		GND pin: Connect all VSS pins to GND (0 V).

Note: All pins with pull-up resistors included, except the $\overline{\text{RESET}}$ pin, can have their pull-up resistors electrically isolated from the pin by software.

3. Functional Description

This section shows the hardware configuration of the TMP95FY64 and explains how it operates.

This device is a version of the created by replacing the predecessor's internal mask ROM with a 256-Kbyte internal flash memory and expanding its internal RAM size to 8 Kbytes. The configuration and the functionality of this device are the same as those of the TMP95CS64. For the functions of this device that are not described here, refer to the TMP95CS64 data sheet.

3.1 Outline of Operation Modes

There are single-chip and single-boot modes. Which mode is selected depends on the device's pin state after a reset.

- Single-chip mode: The device normally operates in this mode. After a reset, the device starts executing the internal flash memory program.
- Single-boot mode: This mode is used to rewrite the internal flash memory by serial transfer (UART). After a reset, the internal boot ROM starts up, executing a on-board rewrite program.

Table 3.1 (1) Operation Mode Setup Table

Operation Mode	Mode Setup Input Pin		
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$	$\overline{\text{EA}}$
Single-chip mode	↑	1	1
Single-boot mode		0	1

Note: The $\overline{\text{BOOT}}$ pin (P30) should be pulled high with an external resistor.

3.2 Memory Map

The memory map of this device differs from that of the TMP95CS64.

Figure 3.2 shows a memory map of the device in single-chip mode and its memory areas that can be accessed in each addressing mode of the CPU.

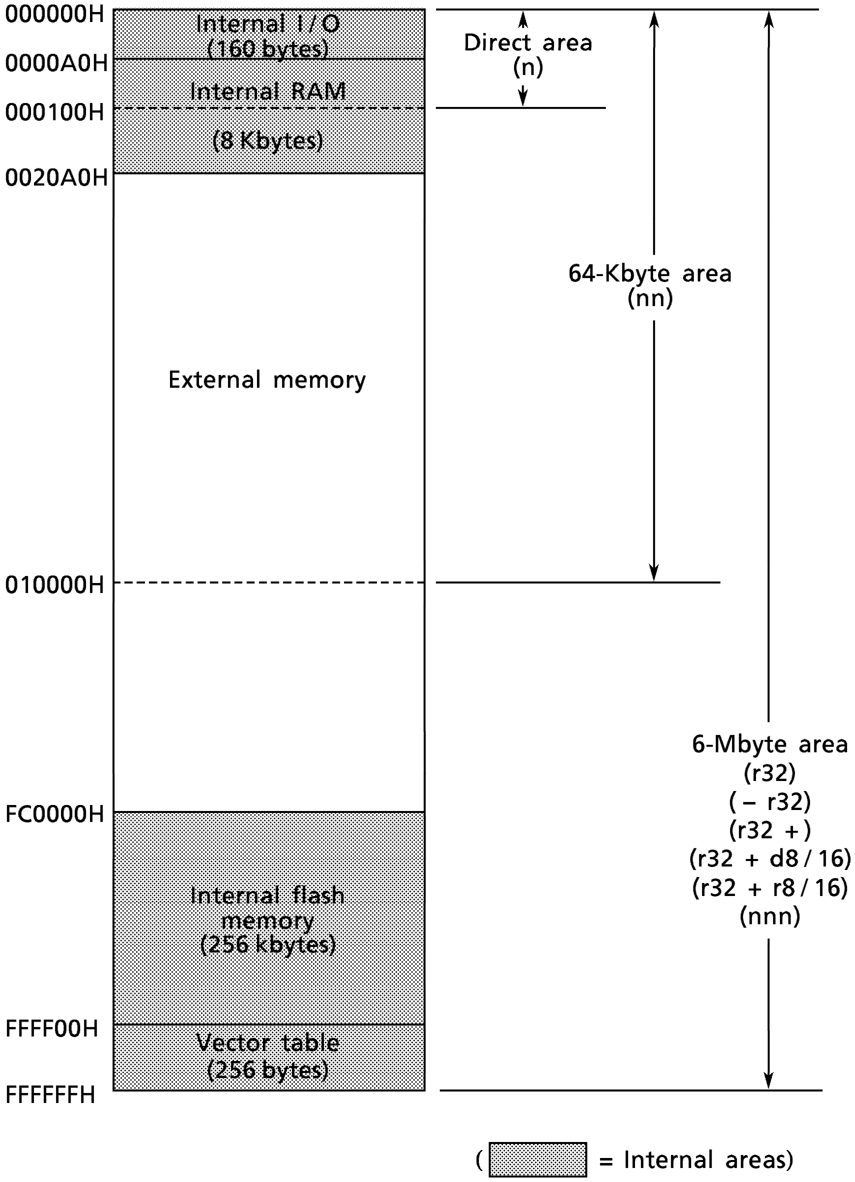


Figure 3.2 Memory Map of the TMP95FY64 (Single-chip Mode)

3.3 Flash Memory

The TMP95FY64 contains an electrically erasable and programmable flash memory using a single 5 V power supply.

The standard JEDEC commands are used to electrically erase and program this flash memory. Once commands are entered, programming and erasure are automatically performed inside the chip. In addition, there are several methods for erasing the flash memory, so that it can be erased the entire chip collectively, one block at a time, or multiple blocks together.

Features:

- Program / erase power supply voltage

  V_{CC} = 5 V ± 10%

- Structure

  256 K × 8 bits /

  128 K × 16 bits (256 Kbytes)

- Functions

  Automatic program

  Automatic erase

  Automatic block erase

  Automatic multiblock erase

  Data polling / toggle bit

- Block erase architecture

  16 Kbytes × 1 / 8 Kbytes × 2 /

  32 Kbytes × 1 / 64 Kbytes × 3

- Mode control

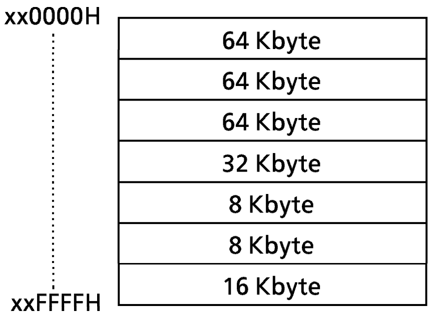
  Based on standard JEDEC commands

- General-purpose flash memory type

  Equivalent to 29F200T

  * Some functions such as block protect are not supported, however.

Block structure:



xx: Depends on the microcomputer's operation mode.

Figure 3.3 (1) Block Structure of the Flash Memory

Command Sequence: Flash memory access by the internal CPU
(Single-boot and user-boot modes)

Command Sequence	Bus Cycles	First Bus Write Cycle		Second Bus Write Cycle		Third Bus Write Cycle		Fourth Bus Read / Write Cycle		Fifth Bus Write Cycle		Sixth Bus Write Cycle	
		Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
Read / reset	1	xXXXXH	F0H	—	—	—	—	—	—	—	—	—	—
Read / reset	3	xAAAAH	AAH	x5554H	55H	xAAAAH	F0H	RA	RD	—	—	—	—
Auto program	4	xAAAAH	AAH	x5554H	55H	xAAAAH	A0H	PA	PD	—	—	—	—
Auto chip erase	6	xAAAAH	AAH	x5554H	55H	xAAAAH	80H	xAAAAH	AAH	x5554H	55H	xAAAAH	10H
Auto block erase	6	xAAAAH	AAH	x5554H	55H	xAAAAH	80H	xAAAAH	AAH	x5554H	55H	BA	30H

The addresses viewed from the CPU side are shown in the table below.

Command Address	CPU Address: A23 to A0																
Addr.	A23 to A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
xXXXXH	Flash memory address area	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
xAAAAH		1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
x5554H		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0

F0H, AAH, 55H, A0H, 80H, 10H, 30H: Command data. Written to DQ7 to DQ0.

RA: Read address
RD: Read data output
] Data is read out in units of bytes / words.

PA: Program address
PD: Program data output
] Data is written to every even address in units of words.

BA: Block address. Each individual block is selected by a combination of A17, A16, A15, A14, and A13.

*: The two reset commands each can reset the device to read mode.

Hardware Sequence Flag List: Flash memory access by the internal CPU

Status		DQ7	DQ6	DQ5	DQ3
Automatic operation under execution	Auto program	DQ7 inverted	Toggle	0	0
	Auto erase (during erase hold time)	0	Toggle	0	0
	Auto erase	0	Toggle	0	1
Time-out (automatic operation failed)	Auto program	DQ7 inverted	Toggle	1	1
	Auto erase	0	Toggle	1	1

Note: DQ8 to DQ15 and DQ0 to DQ2 are "Don't care."

Block Erase Address Table: Flash memory access by the internal CPU

Block	Address in Single Mode					Address Range		Size
	A17	A16	A15	A14	A13	Single Chip	Single Boot	
BA0	L	L	×	×	×	FC0000H-FCFFFFH	010000H-01FFFFH	64 Kbytes
BA1	L	H	×	×	×	FD0000H-FDFFFFH	020000H-02FFFFH	64 Kbytes
BA2	H	L	×	×	×	FE0000H-FEFFFFH	030000H-03FFFFH	64 Kbytes
BA3	H	H	L	×	×	FF0000H-FF7FFFH	040000H-047FFFH	32 Kbytes
BA4	H	H	H	L	L	FF8000H-FF9FFFH	048000H-049FFFH	8 Kbytes
BA5	H	H	H	L	H	FFA000H-FFBFFFH	04A000H-04BFFFH	8 Kbytes
BA6	H	H	H	H	×	FFC000H-FFFFFFH	04C000H-04FFFFH	16 Kbytes

Basic Operation: Flash memory access by the internal CPU

Broadly classified, this flash memory has two operation modes.

These are “Read Mode” in which memory data is read out and “Automatic Operation Mode” in which memory data are automatically erased / rewritten. Automatic operation mode can be entered by executing a command sequence in read mode. No memory data can be read out during automatic operation mode.

(1) Read

To read data from the flash memory, place it in read mode.

Immediately after power-on or when automatic operation has terminated normally, the flash memory goes to read mode. When automatic operation has terminated abnormally or you want read mode to be restored from the other mode, use the reset command that is described later.

(2) Command write

This flash memory uses JEDEC-compliant command control method provided for standard E^2PROMs. Writing to the command register is accomplished by issuing a command sequence to the flash memory. The flash memory latches the entered address and data into the command register as it executes instructions.

To enter command data, use DQ0 to DQ7. Inputs to DQ8 to DQ15 are ignored.

If you want to cancel commands in the middle of a command sequence being entered, issue the reset command. Upon accepting the reset command, the flash memory resets the command register and enters read mode. Also, when an incorrect command sequence is entered, the flash memory resets the command register and enters read mode.

(3) Reset (reset command)

When automatic operation has terminated abnormally, the flash memory does not return to read mode. In this case, use the read / reset command to have the flash memory return to read mode.

Also, if you want to cancel a command in the middle while entering it, you can use the read / reset command. It clears the content of the command register.

(4) Auto program

Write to the flash memory is performed every even address in units of words. In Auto program operation, the program address and program data are latched every even addresses in units of words in the 4th bus write cycle of the command cycle. Upon latching the program data, the flash memory starts auto-programming. Once this operation begins, programming and program verification are automatically performed inside the chip. The status of Auto program operation can be confirmed by checking the hardware sequence flag.

During Auto program operation, command sequences you enter cannot be accepted.

In writing to the flash memory, the cells that contain data “1” can be turned to data “0,” but the cells that contain data “0” cannot be turned to data “1.” To change the data “0” cells to data “1,” you need to perform an erase operation.

If Auto program fails, the flash memory is locked in that mode and does not return to read mode. This status can be confirmed by checking the hardware sequence flag. When in this state, the flash memory needs to be reset by the reset command. Since in this case writing to the address concerned has failed, the memory block that includes this address is faulty. Therefore, make sure this block will not be used.

(5) Auto chip erase

Auto chip erase begins from the 6th bus write cycle of the command cycle ended. Once Auto chip erase starts, all addresses of the flash memory are preprogrammed with data “0,” with the contents then erased and verified for erasure. All this operation is performed automatically inside the chip. The status of Auto chip erase operation can be confirmed by checking the hardware sequence flag.

During Auto chip erase operation, command sequences you enter cannot be accepted.

If Auto chip erase fails, the flash memory is locked in that mode and does not return to read mode. This status can be confirmed by checking the hardware sequence flag. Reset the flash memory by using the reset command. The block in which the failure occurred cannot be detected. Therefore, you need to stop using the device or locate the faulty block by executing block erase. Make sure the faulty block thus found will not be used.

(6) Auto block erase and Auto multiblock erase

Auto block erase begins from the 6th bus write cycle of the command cycle ended after an elapse of the erase hold time. Once Auto block erase starts, all addresses of a selected block are preprogrammed with data "0," with the contents then erased and verified for erasure. All this operation is performed automatically inside the chip. To erase multiple blocks, repeat the 6th bus write cycle and while so doing, enter each block address and the Auto block erase command within the erase hold time. If any other command sequence than Auto block erase is entered during the erase hold time, the flash memory is reset and placed in read mode. The erase hold time is 50 μs, and count starts each time the 6th bus write cycle has ended. The status of Auto block erase operation can be confirmed by checking the hardware sequence flag.

During Auto block erase, command sequences you enter cannot be accepted.

If Auto block erase fails, the flash memory is locked in that mode and does not return to read mode. This status can be confirmed by checking the hardware sequence flag. Reset the flash memory by using the reset command. If multiple blocks have been selected, the block in which the failure occurred cannot be detected. Therefore, you need to stop using the device or locate the faulty block by executing block erase for each block individually. Make sure the faulty block thus found will not be used.

(7) Hardware sequence flags

The hardware sequence flag allows you to confirm the status of the flash memory automatic operation being executed. During automatic operation, data can read from memory at the same timing as in read mode.

When the flash memory finishes automatic operation, it automatically returns to read mode.

The operating status when automatic operation is being executed can be confirmed by checking the hardware sequence flag, and the status after automatic operation is completed can be confirmed by checking whether the data read from memory matches its cell data.

1) DQ7 (DATA polling)

The DATA polling function allows you to confirm the status of the flash memory automatic operation. The DATA polling output begins from the last bus write cycle of the automatic operation command sequence ended. During Auto program operation, the data that has been written to DQ7 is output after being inverted; after the operation is completed, the cell data in DQ7 is output. By reading data out of DQ7, you can identify the operating status. During Auto erase operation, data "0" is output from DQ7; after the operation is completed, data "1" (cell data) is output. If the automatic operation resulted in failure, DQ7 continues outputting the same data that was written to it during automatic operation.

The flash memory frees address latch upon completion of operation, so that when you read data from memory you must enter the address to which data has been written or any block address being erased.

2) DQ6 (toggle bit)

In addition to DATA polling, you can use a toggle bit output function to recognize the status of automatic operation.

Toggle output begins from the last bus write cycle of the automatic operation command sequence ended. This toggle is output to DQ6, with data “1” and “0” output alternately for each read cycle performed. When the automatic operation is completed, DQ6 stops outputting the toggle and instead, outputs its cell data. If the automatic operation has failed, DQ6 continues outputting the toggle.

3) DQ5 (internal timer overtime)

When performing automatic operation normally, the flash memory outputs a “0” to DQ5. If the automatic operation exceeds the flash memory’s internally predetermined time, the DQ5 output changes to a “1.” This means that the automatic operation did not terminate normally, and that the flash memory probably is faulty.

However, when data “1” is written to the data “0” cell, DQ5 outputs a “1,” providing misleading information that the flash memory is faulty. (The flash memory is designed in such a way that although the data “1” cells can be turned to data “0” in program mode, the data “0” cells cannot be turned to data “1.”) In the above case, DQ5 is not showing that the flash memory is faulty, but that the method of command usage is incorrect.

If the automatic operation did not terminate normally, the flash memory is locked and does not return to read mode. Therefore, reset the flash memory using the reset command.

4) DQ3 (block erase timer)

Auto block erase begins from the 6th bus write cycle of the command cycle ended after an elapse of the erase hold time (80 μs). The flash memory outputs a “0” to DQ3 when in the erase hold time and a “1” when it starts erasing. When you want to add a block to be erased, enter it during the block erase hold time. Every time you enter the erase command for each block, the flash memory resets the block erase hold time and starts counting over again. If the automatic operation resulted in failure, DQ3 outputs a “1.”

5) RY / $\overline{\text{BY}}$ (ready / busy)

* This function cannot be used because the flash memory is not connected to the internal CPU.

(8) Flash memory rewrite by the internal CPU

Flash memory rewrite by the internal CPU is accomplished by using the command sequence and hardware sequence flags described above. However, since the built-in flash memory does not read data from its memory cells during automatic operation mode, the rewrite program must be executed external to the flash memory.

There are two methods for flash memory rewrite by the internal CPU. One method uses the single-boot mode prepared in advance; the other method runs the user's original protocol in single-chip mode (user boot).

1) Single boot

In this method, the microcomputer is started in single-boot mode and the flash memory is rewritten using the internal boot ROM program. In this mode, the internal boot ROM is mapped into an area that includes the interrupt vector table, and the boot ROM program is executed in that area. The flash memory is mapped into another address space separately from the boot ROM area. The boot ROM program mainly performs two operations: taking in the rewrite data by serial transfer and rewriting the flash memory. Single boot needs to be performed while interrupts are disabled. Make sure nonmaskable interrupts (e.g., NMI) also are disabled before performing single boot.

For details, refer to Section 3.4, "Single Boot Mode."

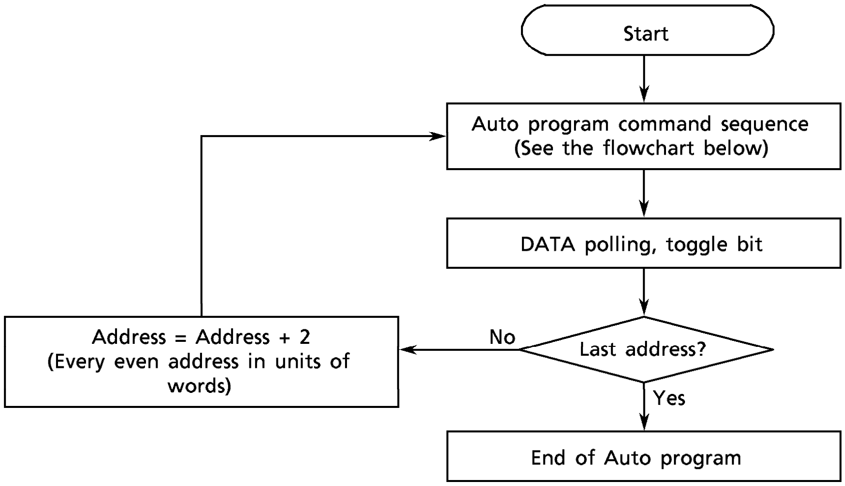
2) User boot

This method runs the user's original flash memory rewrite program. Execute the program in single-chip mode (regular operation mode). In this mode too, the flash memory rewrite program must be executed in another address space separately from that of the flash memory. As in the case of single boot, nonmaskable and all other interrupts must be disabled before performing user boot.

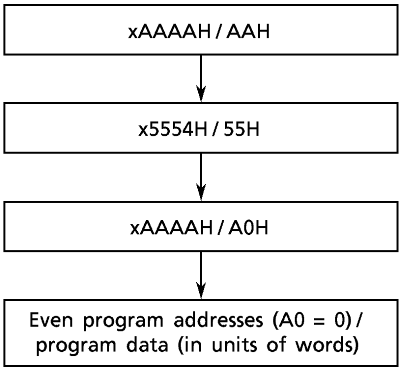
The flash memory rewrite program including routines for taking in the rewrite data and rewriting the flash memory needs to be prepared in advance. When in the main program, switch from regular operation to the flash memory rewrite operation, then execute the flash memory rewrite program you've prepared after expanding it into somewhere outside the flash memory area. For example, you can execute the flash memory rewrite program after expanding it from flash memory into internal RAM or after preparing it in external memory.

Flowchart: Flash memory access by the internal CPU

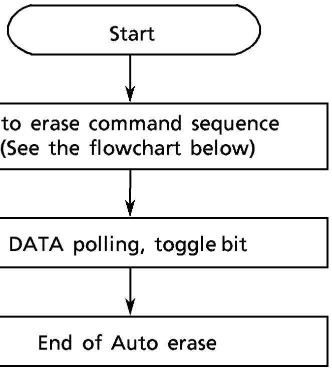
Auto program

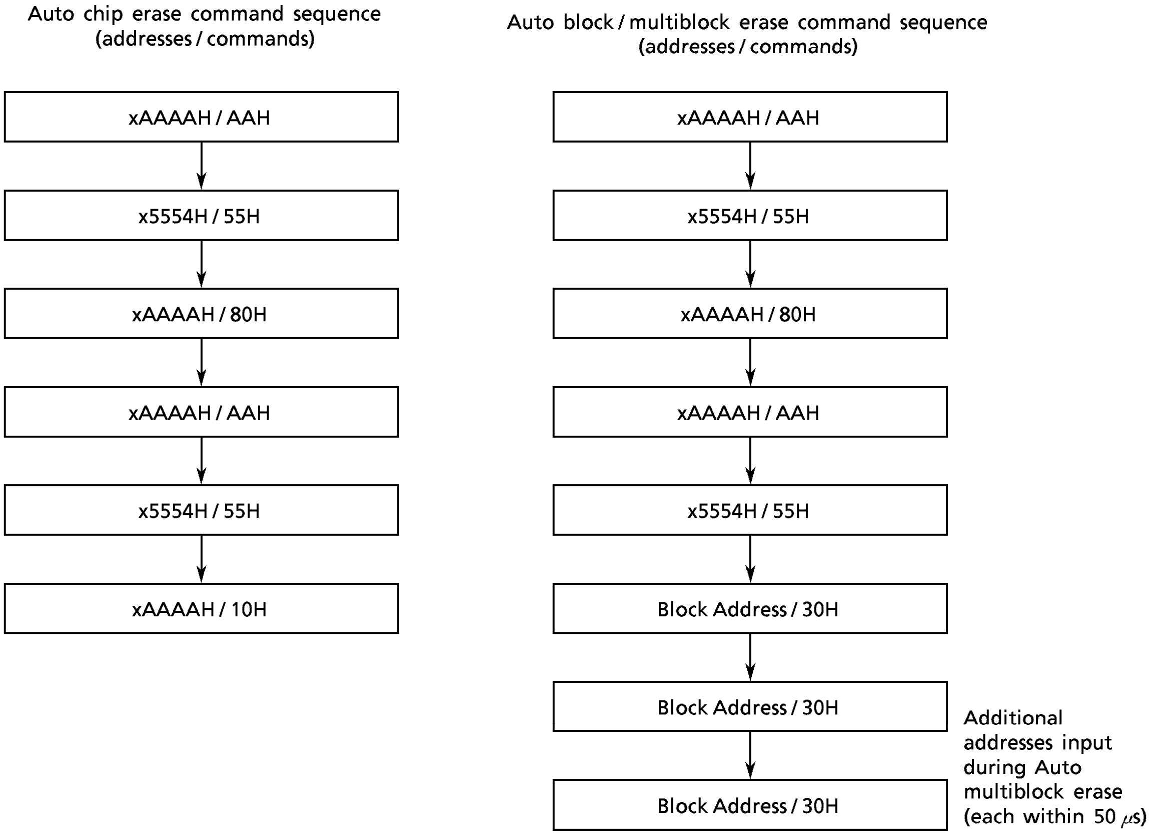

Start
Auto program command sequence (See the flowchart below)
DATA polling, toggle bit
Last address?
No
Address = Address + 2 (Every even address in units of words)
Yes
End of Auto program

Auto program command sequence (addresses / commands)

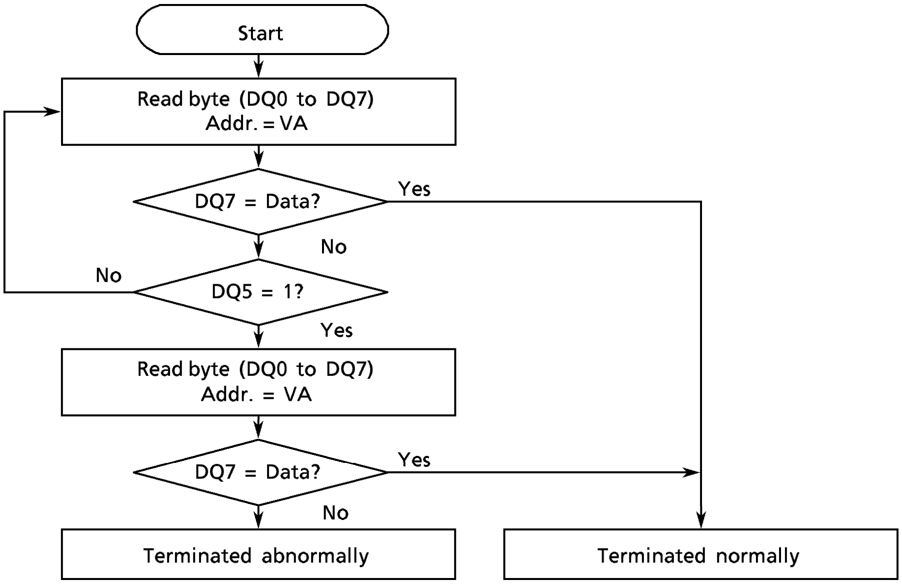

xAAAAH / AAH
x5554H / 55H
xAAAAH / A0H
Even program addresses (A0 = 0) / program data (in units of words)

Auto erase

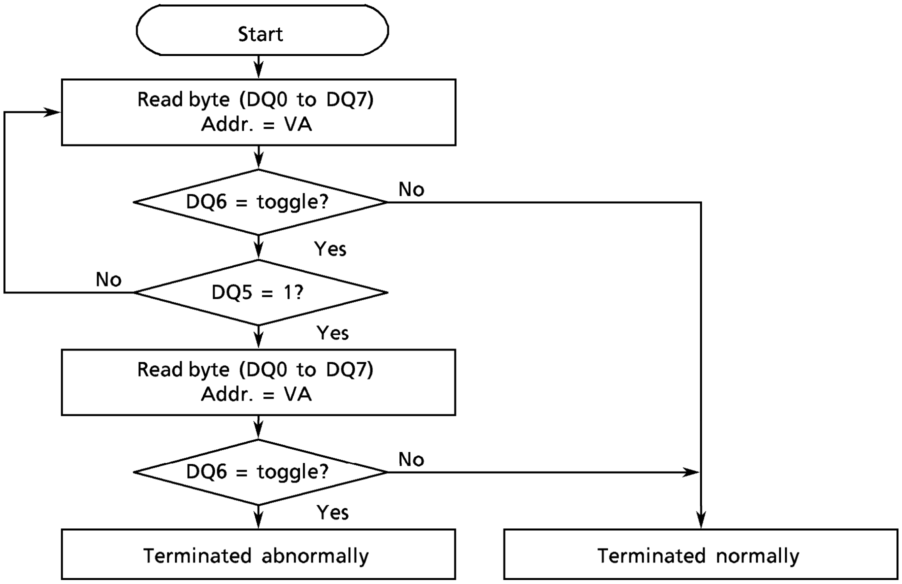

Start
Auto erase command sequence
(See the flowchart below)
DATA polling, toggle bit
End of Auto erase


Auto chip erase command sequence
(addresses / commands)
xAAAAH / AAH
x5554H / 55H
xAAAAH / 80H
xAAAAH / AAH
x5554H / 55H
xAAAAH / 10H
Auto block / multiblock erase command sequence
(addresses / commands)
xAAAAH / AAH
x5554H / 55H
xAAAAH / 80H
xAAAAH / AAH
x5554H / 55H
Block Address / 30H
Block Address / 30H
Block Address / 30H
Additional
addresses input
during Auto
multiblock erase
(each within 50 μs)

DQ7 DATA polling



DQ6 toggle bit



VA: During Auto program, it denotes the address being written to.
During Auto chip erase, it denotes an arbitrary flash memory address.
During Auto block erase, it denotes a selected block address.

3.4 Single Boot Mode

(1) Outline

The TMP95FY64 has single-boot mode available as an on-board programming operation mode. When in single-boot mode, the boot ROM is mapped into memory space. This boot ROM is a mask ROM that contains a program to rewrite the flash memory on-board.

On-board programming is accomplished by first connecting the device's SIO (channel 2) and programming tool (controller) and then sending commands from the controller to the target board.

The boot program included in the boot ROM also has the function of a loader, so it can transfer program data from an external source into the device's internal RAM.

Figure 3.4 (1) shows an example of how to connect the programming controller and the target board.

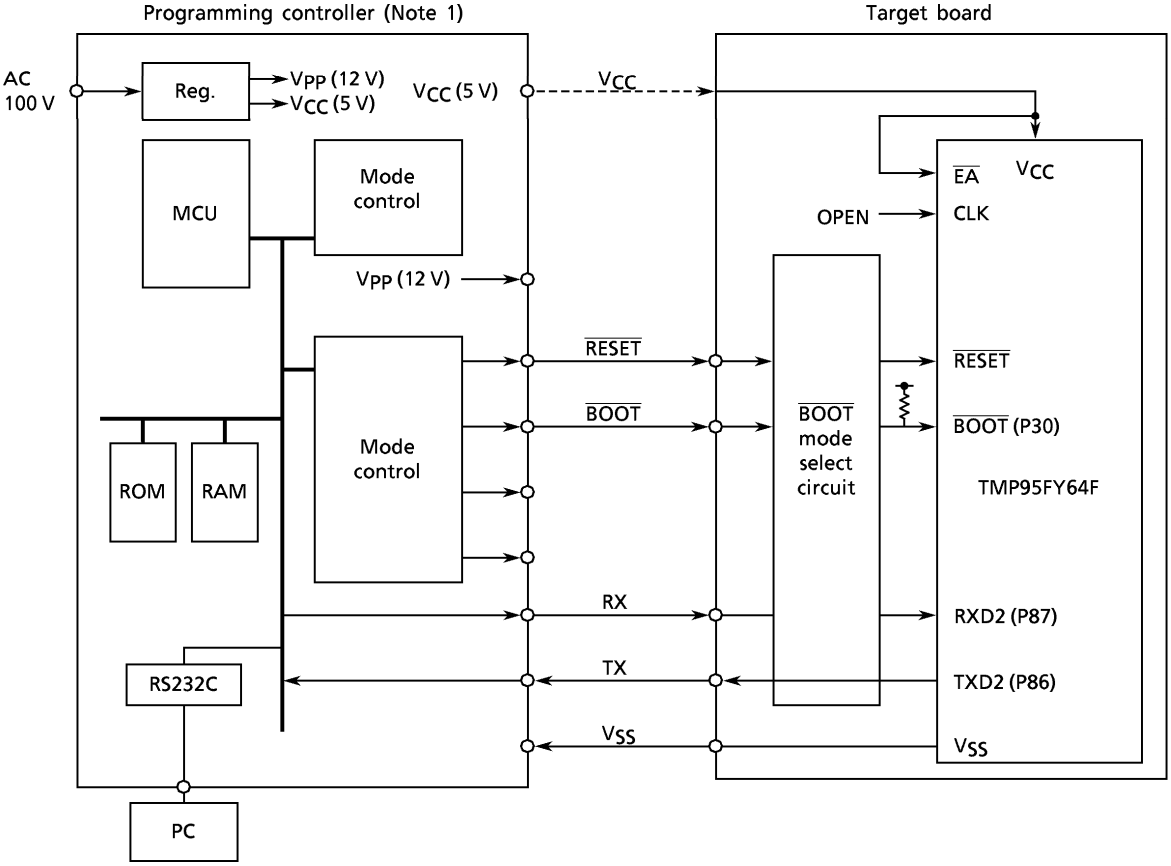


Figure 3.4 (1) Example for Connecting Units for On-board Programming

Note: One of the programming controllers supported for the TMP95FY64 is the AF200 (Advanced On-board Flash Microcomputer Programmer) from Yokogawa Digital Computer Co. For details, refer to the manual included with the AF200.

Where to contact: Yokogawa Digital Computer Co.

Equipment Operations Center, Microcomputer System Headquarters

TEL: 0423-33-6224

(2) Mode settings

To execute on-board programming, start the TMP95FY64 in single-boot mode. Settings necessary to start up in single-boot mode are shown below.

$\overline{\text{EA}}$	=	H
$\overline{\text{BOOT}}$ (P30)	=	L
$\overline{\text{RESET}}$	=	↑

After setting the $\overline{\text{EA}}$ and $\overline{\text{BOOT}}$ pins each to the above conditions, drive the signal input to the $\overline{\text{RESET}}$ pin high. The TMP95FY64 starts up in single-boot mode.

(3) Memory map

Figure 3.4 (2) compares memory maps in single-chip and single-boot modes. When in single boot mode, the internal flash memory is mapped into addresses 10000H through 4FFFFH, as shown here.

You'll also find that the boot ROM (MROM) is mapped into addresses FFF800H through FFFFFFH.

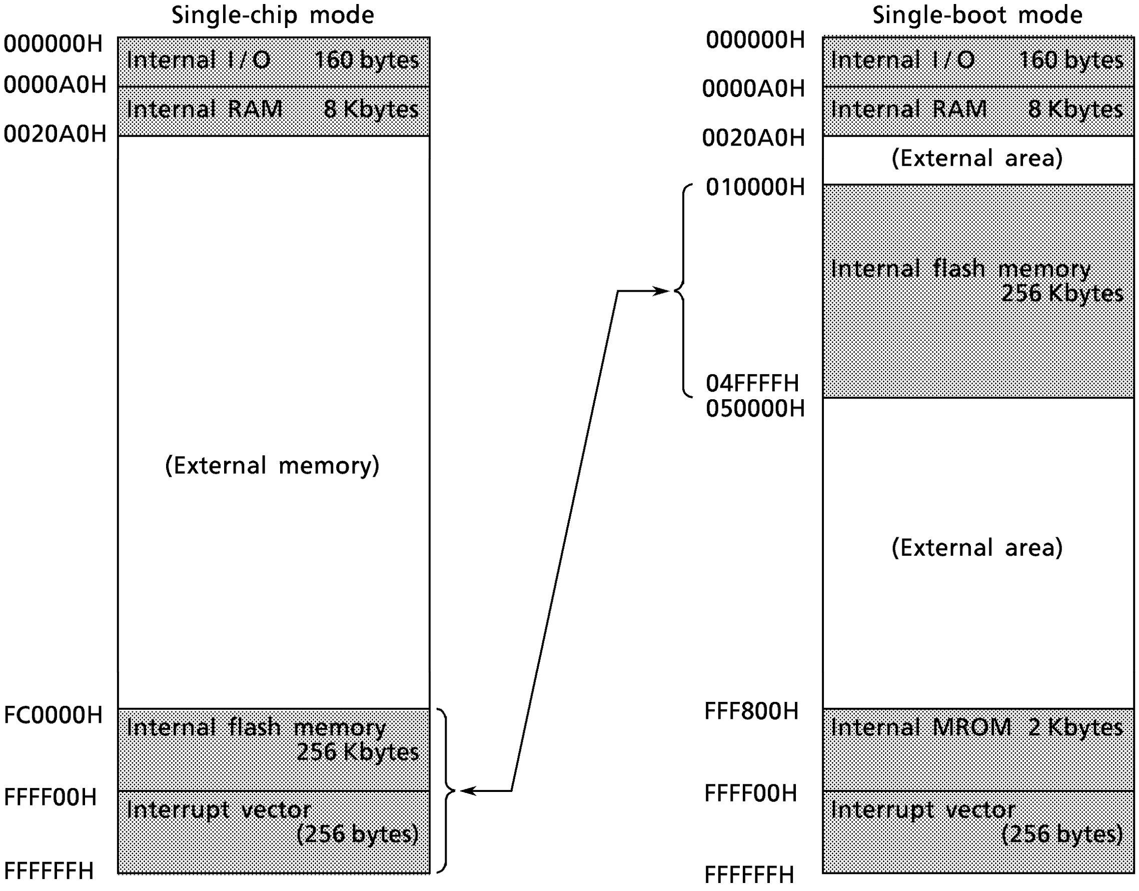


Figure 3.4 (2) Comparison of Memory Maps

(4) Interface specifications

The following shows the SIO communication format used in single-boot mode.

Before on-board programming can be executed, the communication format on the programming controller side must also be set up in the same way as for the TMP95FY64.

Note that although the default baud rate is 9,766 bps (@fc = 25 MHz), it can be changed to other values as shown in Table 3.4 (1).

Communication channel: SIO channel 2

Serial transfer mode: UART (asynchronous communication) mode, full-duplex communication

Data length: 8 bits

Parity it: None

Stop bit: 1 bit

Baud rate (default): 9,766 bps (@fc = 25 MHz)

(5) Data transfer format

Tables 3.4 (1) through 3.4 (7) show baud rate modification data, operation commands, and data transfer format in each operation mode, respectively.

Also refer to the description of boot program operation in the latter pages of this manual as you read these tables.

Table 3.4 (1) Baud Rate Modification Data (@fc = 25 MHz)

Baud Rate Modification Data	04H	05H	06H	07H	0AH	18H	28H
Baud Rate (bps)	78125	65104	55804	39063	32552	19531	9766

Note: The baud rates currently supported by the AF200 are 9600, 19200, 31250, and 62500 bps only.

Table 3.4 (2) Operation Command Data

Operation Command Data	Operation Mode
30H	Flash memory rewrite
60H	RAM loader
90H	Flash memory SUM

Table 3.4 (3) Operating frequency and baud rate in Single Boot mode.: TMP95FY64

Reference baud rate (bps)		9600		19200		31250		38400		57600		62500		76800	
Baud rate change data		28h		18h		0Ah		07h		06h		05h		04h	
Ref. Xtal (MHz)	Area (MHz)	Baud rate (bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)
2.4576	2.44 to 2.48	9600	0	19200	0			38400	0						
3	2.97 to 3.03	9375	-2.34												
3.6864	3.64 to 3.74	9600	0	19200	0					57600	0				
4.9152	4.85 to 5.07	9600	0	19200	0			38400	0					76800	0
5		9766	+ 1.73	19531	+ 1.72			39063	+ 1.73					78125	+ 1.73
6	5.91 to 6.23	9375	-2.34	18750	-2.34	31250	0								
6.144		9600	0	19200	0	32000	+ 2.4								
7.3728	7.26 to 7.48	9600	0	19200	0			38400	0	57600	0				
8	7.84 to 8.16	9615	+ 0.16			31250	0					62500	0		
9.8304	9.64 to t10.20	9600	0	19200	0	30720	-1.7	38400	0					76800	0
10		9766	+ 1.73	19531	+ 1.72	31250	0	39063	+ 1.73					78125	+ 1.73
12	11.76 to 12.75	9375	-2.34	18750	-2.34	31250	0	37500	-2.34			62500	0		
12.288		9600	0	19200	0	32000	+ 2.4	38400	0			64000	+ 2.4		
12.5		9766	+ 1.73	19531	+ 1.72	32552	+ 4.17	39063	+ 1.73			65104	+ 4.17		
14.7456	14.46 to 15.04	9600	0	19200	0	32914	+ 5.3	38400	0	57600	0			76800	0
16	15.68 to 16.32	9615	+ 0.16	19231	+ 0.16	31250	0					62500	0		
18	17.64 to 18.36	9375	-2.34	18750	-2.34	31250	0			56250	-2.34				
19.6608	19.27 to 20.40	9600	0	19200	0	30720	-1.7	38400	0			61440	-1.7	76800	0
20		9766	+ 1.73	19531	+ 1.72	31250	0	39063	+ 1.73			62500	0	78125	+ 1.73
21.18	20.76 to 22.56	9193	-4.24	18385	-4.24	30085	-3.73	36771	-4.24	55156	-4.24				
22.1184		9600	0	19200	0	31418	+ 0.54	38400	0	57600	0				
24.5760	24.09 to 25.50	9600	0	19200	0	32000	+ 2.4	38400	0	54857	-4.76	64000	+ 2.4	76800	0
25		9766	+ 1.73	19531	+ 1.72	32552	+ 4.17	39063	1.73	55804	-3.12	65104	+ 4.7	78125	+ 1.73
26.88	26.35 to 27.54	9545	-0.57	19091	-0.57	30000	-4	38182	-0.57						
27		9588	-0.13	19176	-0.13	30134	-3.57	38352	-0.13						
32	31.36 to 32.64	9615	+ 0.16	19231	+ 0.16	31250	0	38462	+ 0.16	55556	-3.55	62500	0		

Reference frequency: High speed oscillator frequency supported in Single boot mode.
When the Single boot mode is used for programming Flash memory, each of reference frequency should be used.
Area: Clock frequency area detected for reference frequency. The Single boot would not be executed at the others frequency.

Note: The Auto-detection of MCU operating frequency will be normally done when the total error between transmit baud rate (9600 bps) of program controller, oscillator frequency and detecting of matching data is under + / −3%.

Table 3.4 (4) Boot Program Transfer Format (@fc = 25 MHz) (For Flash Memory Rewrite)

	Number of Bytes Transferred	Transfer Data from Controller to TMP95FY64	Baud Rate	Transfer Data from TMP95FY64 to Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) –	9766 bps 9766 bps	– (Baud rate auto set) OK: Echoback data (5AH) NG: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 3.16 (1).) –	9766 bps 9766 bps	– OK: Echoback data NG: A1H × 3, A2H × 3, A3H × 3
	5th byte 6th byte	Operation command data (30AH) –	Changed new baud rate Changed new baud rate	– OK: Echoback data (30H) NG: A1H × 3, A2H × 3, A3H × 3, 63H × 3
	7th byte	–	Changed new baud rate	OK: C1H NG: 64H × 3
	8th byte : n'th – 2 byte	Extended Intel Hex format (binary)	Changed new baud rate	–
	n'th – 1 byte	–	Changed new baud rate	OK: SUM (High) NG: Nothing transmitted
	n'th byte	–	Changed new baud rate	OK: SUM (Low) NG: Nothing transmitted
	n'th + 1 byte	(Wait for the next operation command data)	Changed new baud rate	–

*1: "xxHX3" denotes that operation stops after sending 3 bytes of xxH.
*2: Refer to "Notes on Extended Intel Hex Format (Binary)" in the latter page of this manual.
*3: Refer to "Notes on SUM" in the latter page of this manual.

Table 3.4 (5) Boot Program Transfer Format (@fc = 25 MHz) (For RAM Loader)

	Number of Bytes Transferred	Transfer Data from Controller to TMP95FY64	Baud Rate	Transfer Data from TMP95FY64 to Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) –	9766 bps 9766 bps	– (Baud rate auto set) OK: Echoback data (5AH) NG: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 3.16 (1).) –	9766 bps 9766 bps	– OK: Echoback data NG: A1H × 3, A2H × 3, A3H × 3, 62H × 3
	5th byte 6th byte	Operation command data (60H)	Changed new baud rate Changed new baud rate	– OK: Echoback data (60H) NG: A1H × 3, A2H × 3, A3H × 3, 63H × 3
	7th byte 8th byte	Address 23-16*2 in which to store Password count –	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted NG: A1H × 3, A2H × 3, A3H × 3
	9th byte 10th byte	Address 15-08*2 in which to store Password count –	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted NG: A1H × 3, A2H × 3, A3H × 3
	11th byte 12th byte	Address 07-00*2 in which to store Password count –	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted NG: A1H × 3, A2H × 3, A3H × 3
	13th byte 14th byte	Address 23-16*2 at which to start Password comparison –	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted NG: A1H × 3, A2H × 3, A3H × 3
	15th byte 16th byte	Address 15-08*2 at which to start Password comparison –	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted NG: A1H × 3, A2H × 3, A3H × 3
	17th byte 18th byte	Address 07-00*2 at which to start Password comparison –	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted NG: A1H × 3, A2H × 3, A3H × 3
	19th byte : m'th byte	Password string –	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted NG: A1H × 3, A2H × 3, A3H × 3
	m'th + 1 byte : n'th – 2 byte	Extended Intel Hex format (binary)		–
	n'th – 1 byte	–	Changed new baud rate	OK: SUM (High) NG: Nothing transmitted
	n'th byte	–	Changed new baud rate	OK: SUM (Low) NG: Nothing transmitted
RAM	–	Jump to the user program's start address		

*1: "xxHX3" denotes that operation stops after sending 3 bytes of xxH.
*2: Refer to "Notes on Password" in the latter page of this manual.
*3: Refer to "Notes on Extended Intel Hex Format (Binary)" in the latter page of this manual.
*4: Refer to "Notes on SUM" in the latter page of this manual.

Table 3.4 (6) Boot Program Transfer Format (@fc = 25 MHz) (For Flash Memory SUM)

	Number of Bytes Transferred	Transfer Data from Controller to TMP95FY64	Baud Rate	Transfer Data from TMP95FY64 to Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) –	9766 bps 9766 bps	– (Baud rate auto set) OK: Echoback data (5AH) NG: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 3.16 (1).) –	9766 bps 9766 bps	– OK: Echoback data NG: A1H × 3, A2H × 3, A3H × 3, 62H × 3
	5th byte 6th byte	Operation command data (90H) –	Changed new baud rate Changed new baud rate	– OK: Echoback data (90H) NG: A1H × 3, A2H × 3, A3H × 3, 63H × 3
	7th byte	–	Changed new baud rate	OK: SUM (High) NG: –
	8th byte	–	Changed new baud rate	OK: SUM (Low) NG: –
	9th byte	(Wait for the next operation command data)	Changed new baud rate	–

*1: "xxHX3" denotes that operation stops after sending 3 bytes of xxH.
*2: Refer to "Notes on SUM."

(6) Description of boot program operation

When you start the TMP95FY64 in single-boot mode, the boot program starts up. The boot program provides the functions described below.

For details about these functions, refer to ① Flash memory rewrite program through ③ Flash memory SUM command in the pages that follow.

1. Flash memory rewrite

   The flash memory is erased the entire chip (256 Kbytes) collectively. Then data are written to the specified flash memory addresses. The controller should send the write data in the Extended Intel Hex format (binary).

   If no errors are encountered till the end record, the SUM of 256 Kbytes of flash memory is calculated and the result is returned to the controller.

2. RAM loader

   The RAM loader transfers the data into the internal RAM that has been sent from the controller in Extended Intel Hex format. When the transfer has terminated normally, the RAM loader calculates the SUM and sends the result to the controller before it starts executing the user program. The execution start address is the first address received. This RAM loader function provides the user's own way to control on-board programming.

   To execute on-board programming in the user program, you need to issue the flash memory command sequence described in the preceding section of this manual. (Must be matched to the flash memory addresses in single-boot mode.)

   The RAM loader command checks the result of password collation prior to program execution. If the passwords did not match, the program is not executed.

3. Flash memory SUM

   The SUM of 256 Kbytes of flash memory is calculated and the result is returned to the controller.

   The boot program does not support the operation commands to read data from the flash memory. Instead, it has this SUM command to use. By reading the SUM, it is possible to manage Revisions of application programs.

   ① Flash memory rewrite command (Table 3.4 (4))

   1. The receive data in the first byte is the matching data. When the boot program starts in single-boot mode, it goes to a state in which it waits for the matching data to receive. Upon receiving the matching data, it automatically adjusts the serial channels' initial baud rate to 9,766 bps.

      The matching data is 5AH.

   2. The 2nd byte is used to echo back 5AH to the controller upon completion of the automatic baud rate setting in the first byte. If the device fails in automatic baud rate setting, it goes to an idle state.

3. The receive data in the 3rd byte is the baud rate modification data. The seven kinds of baud rate modification data shown in Table 3.4 (1) are available. Even when you do not change the baud rate, be sure to send the initial baud rate data (28h: 9,766 bps @fc = 25 MHz).

   Baud rate modification becomes effective after the echoback transmission is completed.

4. The 4th byte is used to echo back the received data to the controller when the data received in the third byte is one of the baud rate modification data corresponding to the device's operating frequency. Then the baud rate is changed. If the received baud rate data does not correspond to the device's operating frequency, the device goes to an idle state after sending 3 bytes of baud rate modification error code (62H).

5. The receive data in the 5th byte is the command data (30H) to rewrite the flash memory.

6. The 6th byte is used to echo back the received data (in this case, 30H) to the controller when the data received in the 5th byte is one of the operation command data in Table 3.4 (2). And the flash memory rewrite routine is called. If the received data is none of the operation command data, the device goes to an idle state after sending 3 bytes of operation command error code (63H).

7. The transmit data in the 7th byte indicates whether collective erase (256 Kbytes) has terminated normally. When collective erase (256 Kbytes) has terminated normally, the device returns collective erase terminated normally code (C1H) to the controller.

   If an erase error occurs, the device goes to an idle state after returning three bytes of erase error code (64H) to the controller.

   The controller should send the next data to the device after receiving the collective erase terminated normally code (C1H).

8. The receive data in the 8th byte through n'th − 2 byte are received as binary data in Extended Intel Hex format. No received data are echoed back to the controller.

   The flash memory rewrite routine ignores the received data until it receives the start mark (3AH for ":") in Extended Intel Hex format. Nor does it send error code to the controller. After receiving the start mark, the routine receives a range of data from data length to checksum and writes the received write data to the specified flash memory addresses successively. Since bits 23 to 16 of the address pointer during write are by default 00H, the first record type must always be an extended record.

   After receiving one record of data from start mark to checksum, the routine goes to a start mark waiting state again.

   If a write error, receive error, or Extended Intel Hex format error occurs, the device goes to an idle state without returning error code to the controller.

   Because the flash memory rewrite routine executes a SUM calculation routine upon detecting the end record, the controller should be placed in a SUM waiting state after sending the end record to the device.

9. The n'th − 1 and the n'th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to "Notes on SUM" in the latter page of this manual. The SUM calculation is performed only when no write error, receive error, or Extended Intel Hex format error has been encountered after detecting the end record. The time required to calculate the SUM of the 256 Kbytes of flash memory area is approximately 400 ms at fc = 20 MHz. After SUM calculation, the device sends the SUM data to the controller. The controller should determine whether writing to the flash memory has terminated normally depending on whether the SUM value is received after sending the end record to the device.

10. The receive data in the n'th + 1 byte, if rewriting terminated normally, places the device in a state waiting for the next operation command data.

② RAM loader command (Table 3.4 (5))

1. The transmit / receive data in the 1st through the 4th bytes are the same as in the case of flash memory rewrite commands.

2. The receive data in the 5th byte is the RAM loader command data (60H).

3. The 6th byte is used to echo back the received data (in this case, 60H) to the controller when the data received in the 5th byte is one of the operation command data in Table 3.4 (2). Then the RAM loader routine is called. If the received data is none of the operation command data, the device goes to an idle state after returning three bytes of operation command error code (63H) to the controller.

4. The receive data in the 7th byte is the data for bits 23 to 16 of the address in which the password count is stored. Three bytes of password count storage address are required. The data indicated by this address is the password count. Note that if the password count is equal to or less than 8, the command is canceled.

5. Nothing is sent in the 8th byte to the controller when the data received in the 7th byte has no error. If a receive error is encountered, the device goes to an idle state after returning three bytes of relevant error code to the controller.

6. The 9th through the 12th bytes respectively are bits 15 to 8 and bits 7 to 0 of the password count storage address and are the data used when a receive error is encountered to return error code to the controller. For these operations, refer to paragraphs 4 and 5 above.

7. The receive data in the 13th byte are bits 23 to 16 of the address at which the password comparison is started. Three bytes of password comparison start address are required. Passwords are compared beginning with this address.

8. Nothing is sent in the 14th byte to the controller when the data received in the 13th byte has no error. If a receive error is encountered, the device goes to an idle state after returning three bytes of relevant error code to the controller.

9. The 15th through the 18th bytes respectively are bits 15 to 8 and bits 7 to 0 of the password comparison start address and are the data returned to the controller. For these operations, refer to paragraphs 7 and 8 above.

10. The 19th through the m'th bytes are the password data. The number of passwords or the password count is the data (N) indicated by the password count storage address. The password data are compared for N entries beginning with the password comparison start address. The controller should send N bytes of password data to the device. If the passwords do not match, the device goes to an idle state without returning error code to the controller.

11. The receive data in the m’th + 1 through the n’th – 2 bytes are received as binary data in Extended Intel Hex format. No received data are echoed back to the controller.

    The RAM loader routine ignores the received data until it receives the start mark (3AH for “:”) in Extended Intel Hex format. Nor does it send error code to the controller. After receiving the start mark, the routine receives a range of data from data length to checksum.

    The received write data are successively written to the specified flash memory addresses. Since bits 23 to 16 of the address pointer during write are by default 00H, the first record type does not always have to be an extended record.

    After receiving one record of data from start mark to checksum, the routine goes to a start mark waiting state again.

    If a receive error or Extended Intel Hex format error occurs, the device goes to an idle state without returning nothing to the controller.

    Because the RAM loader routine executes a SUM calculation routine upon detecting the end record, the controller should be placed in a SUM waiting state after sending the end record to the device.

12. The n’th – 1 and the n’th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to “Notes on SUM” in the latter page of this manual. The SUM calculation is performed only when no receive error or Extended Intel Hex format error has been encountered after detecting the end record. The time required to calculate the SUM is approximately proportional to the number of data written to RAM. The time required to calculate the SUM of a 4 Kbytes of RAM area, for example, is approximately 6 ms at fc = 20 MHz. After SUM calculation, the device sends the SUM data to the controller. The controller should determine whether writing to RAM has terminated normally depending on whether the SUM value is received after sending the end record to the device.

13. The boot program jumps to the first address that is received as data in Extended Intel Hex format after sending the SUM to the controller.

③ Flash memory SUM command (Table 3.4 (6))

1. The transmit / receive data in the 1st through the 4th bytes are the same as in the case of flash memory rewrite commands.

2. The receive data in the 5th byte is the flash memory SUM command data (90H).

3. The 6th byte is used to echo back the received data (in this case, 90H) to the controller when the data received in the 5th byte is one of the operation command data in Table 3.4 (2). Then the flash memory SUM processing routine is called. If the received data is none of the operation command data, the device goes to an idle state after returning three bytes of operation command error code (63H) to the controller.

4. The 7th and the 8th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to "Notes on SUM" in the latter page of this manual.

5. The receive data in the 9th byte places the device in a state waiting for the next operation command data.

④ Boot program transmit data

The boot program sends the processing status to the controller using various code. The transmit data (processing code) are listed in the table below.

Table 3.4 (7) Boot Program Transmit Data

Transmit Data	Meaning of Transmit Data
C1H	Collective erase of flash memory chip terminated normally.
62H, 62H, 62H	Baud rate modification error occurred.
63H, 63H, 63H	Operation command error occurred.
64H, 64H, 64H	Flash memory erase error occurred.
A1H, A1H, A1H	Framing error in received data occurred.
A2H, A2H, A2H	Parity error in received data occurred.
A3H, A3H, A3H	Overrun error in received data occurred.

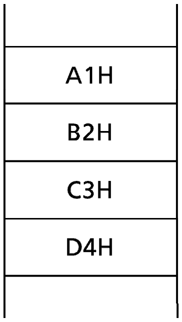
*1: When this receive error occurs when receiving data in Extended Intel Hex format, the device does not send the receive error code to the controller.

⑤ Notes on SUM

1. Calculation method

SUM consists of byte + byte······+ byte, the sum of which is returned in word as the result. Namely, data is read out in byte and sum of which is calculated, with the result returned in word.

Example:



If the data to be calculated consists of the four bytes shown to the left, SUM of the data is

A1H + B2H + C3H + D4H = 02EAH

∴ SUM (HIGH) = 02H
SUM (LOW) = EAH

The SUM returned when executing the flash memory rewrite command, RAM loader command, or flash memory SUM command is calculated in the manner shown above.

2. Calculation data

The data from which SUM is calculated are listed in Table 3.4 (8) below.

Table 3.4 (8) SUM Calculation Data

Operation Mode	Calculation Data	Remarks
Flash memory rewrite command	Data in the entire area (256 Kbytes) of flash memory	The received flash memory or RAM write data is not the only data to be calculated for SUM. Even when the received addresses are noncontiguous and there are some unwritten areas, data in the entire memory area is calculated.
RAM loader command	Data written in an area ranging from the first address received to the last address received	
Flash memory SUM command	Data in the entire area (256 Kbytes) of flash memory	–

⑥ Notes on Extended Intel Hex Format (binary)

1. For the flash memory rewrite command, always make sure the first record type is an extended record. This is because the internal flash memory of the TMP95FY64 is located in a memory space starting from address 10000H, so that bits 23 to 16 of the address pointer when writing to the flash memory are, by default, 00H.

2. For the RAM loader command, the first record type does not always have to be an extended record. This is because bits 23 to 16 of the address pointer when writing to the flash memory are, by default, 00H.

3. After receiving the checksum of a record, the device waits for the start mark (3AH for ":") of the next record. Therefore, the device ignores all data received between records during that time unless the data is 3AH.

4. Make sure that once the controller program has finished sending the checksum of the end record, it does not send anything and waits for two bytes of data to be received (upper and lower bytes of SUM). This is because after receiving the checksum of the end record, the boot program calculates the SUM and returns the calculated SUM in two bytes to the controller.

5. If a write error (for only the flash memory rewrite command), receive error, or Extended Intel Hex format error occurs, the device goes to an idle state without returning error code to the controller. In the following cases, an Extended Intel Hex format error is assumed:

   - When TYPE is not 00H, 01H, or 02H
   - When a checksum error occurred
   - When the data length of an extended record (TYPE = 02H) is not 02H
   - When the address of an extended record (TYPE = 02H) is not 0000H
   - When the data in the 2nd byte of an extended record (TYPE = 02H) is not 00H
   - When the data length of the end record (TYPE = 01H) is not 00H
   - When the address of the end record (TYPE = 01H) is not 0000H

Example: When writing to an area from address 1FFF8H to address 2002FH, the transfer format should be like the one shown in Table 3.4 (9).

Table 3.4 (9) Example of Transfer Format for Flash Memory Rewrite Command

Direction of Data	Meaning of Data Extended Intel Hex Format (n'th − 2 byte in item 8 of Table 3.4 (4))	Data
Controller to TMP95FY64	Extended record	: 02 0000 02 1000 EC zz
Controller to TMP95FY64	Data record (data length: 08H)	: 08 FFF8 00 xxxxxx CS zz
Controller to TMP95FY64	Extended record	: 02 0000 02 2000 DC zz
Controller to TMP95FY64	Data record (data length: 30H)	: 30 0000 00 yyyyyyyy CS zz
Controller to TMP95FY64	End record	: 00 0000 01 FF ww
TMP95FY64 to controller	SUM (upper byte) (n'th − 1 byte in Table 3.4 (4))	SUM (upper byte)
TMP95FY64 to controller	SUM (lower byte) (n'th byte in Table 3.4 (4))	SUM (lower byte)
Controller to TMP95FY64	Operation command (n'th + 1 byte in Table 3.4 (4))	Next operation command data

Note: The colon ":" denotes the start mark (3AH).
xx, yy denote the data written to flash memory.
CS, EC, DC, FF denote the checksum data.
zz denotes the data that can be sent by the controller without causing a problem.
ww denotes the data that cannot be sent by the controller.

⑦ Notes on Passwords

The area in which passwords can be specified is located at addresses 12000H to 4DFFFH. Figure 3.4 (3) schematically shows the password area.

1. Password count storage address (PNSA)

The content of the address specified by PNSA is the password count (N). In the following cases, a password error is assumed:

- PNSA < address 12000H
- Address 4DFFFH < PNSA
- N < 8

2. Password comparison start address (PCSA)

The passwords are compared beginning with the address specified by PCSA. The specified password area is from PCSA to PCSA + N. In the following cases, a password error is assumed:

- PCSA < address 12000H
- Address 4DFFFH < PCSA + N − 1
- When the specified password area contains three or more consecutive bytes of the same data. However, if all data in the vector part (4FF00H to 4FFFFH) are FFH, the device is assumed to be a blank product, in which no check is made of the passwords.

3. Password string

A string of passwords in the received data are compared with the data in the flash memory. In the following cases, a password error is assumed:

- When the received data does not match the data in the flash memory

4. Handling of password error

When a password error occurs, the device goes to an idle state.

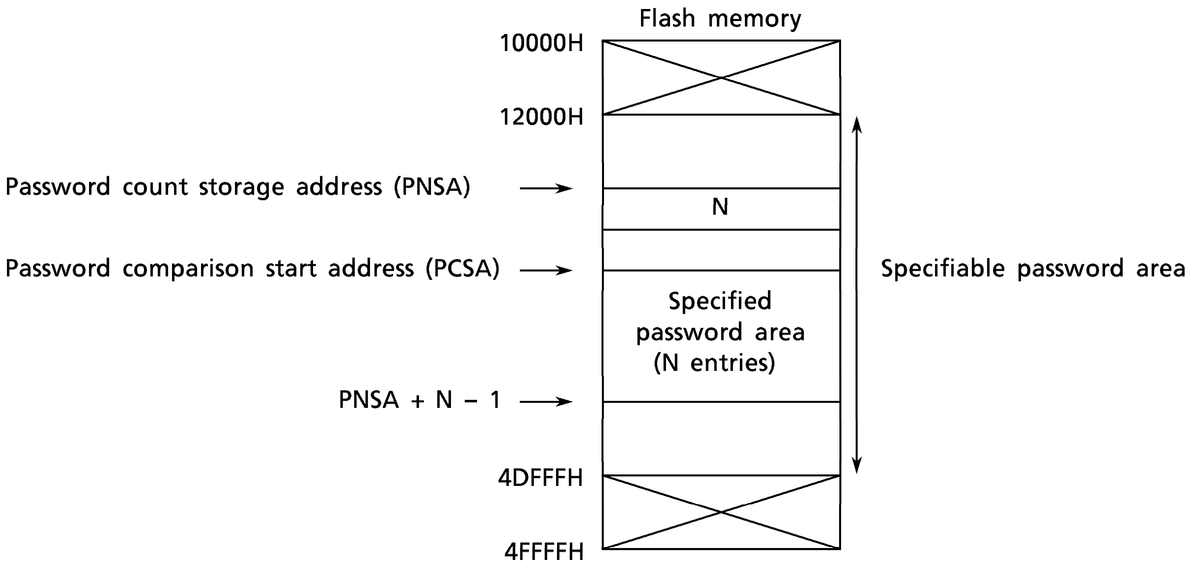
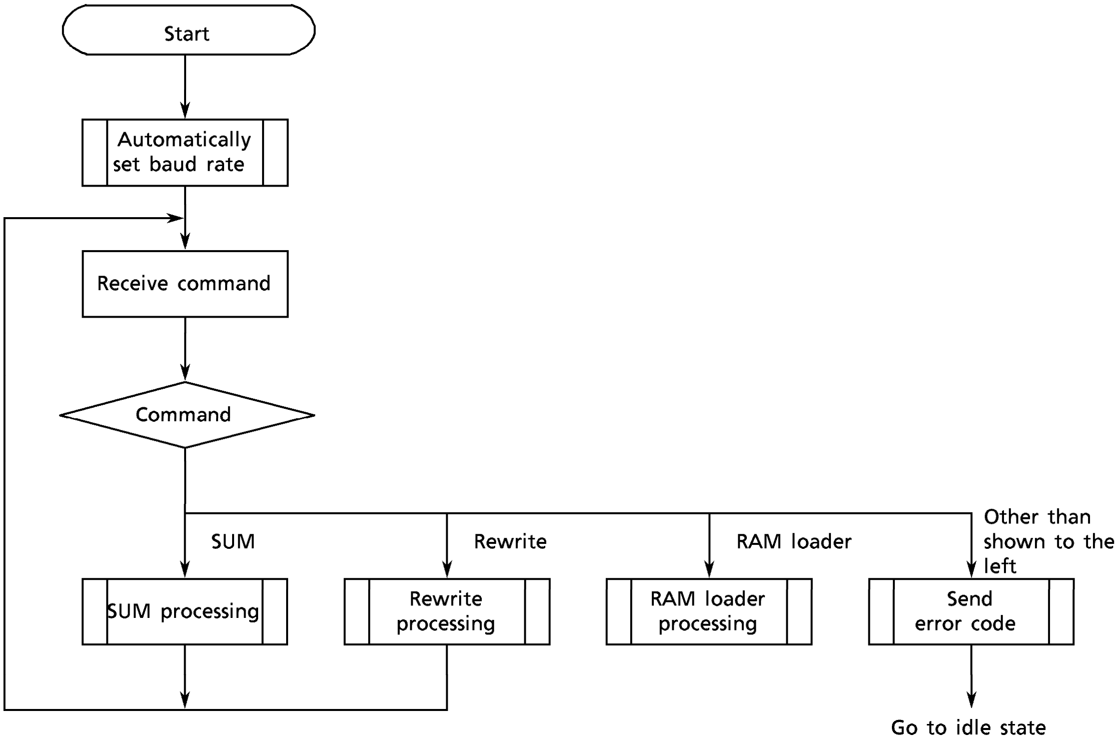
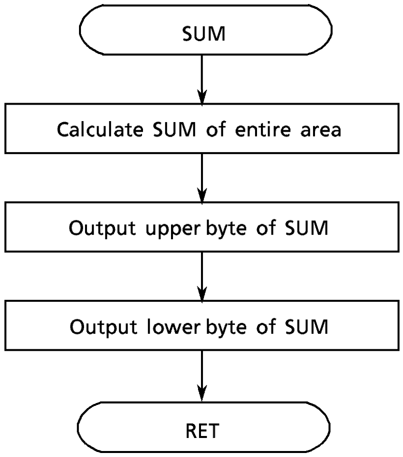

Flash memory
10000H
12000H
Password count storage address (PNSA)
N
Password comparison start address (PCSA)
Specified
password area
(N entries)
PNSA + N − 1
4DFFFH
4FFFFH
Specifiable password area

Figure 3.4 (3) Conceptual Diagram of a Password Area

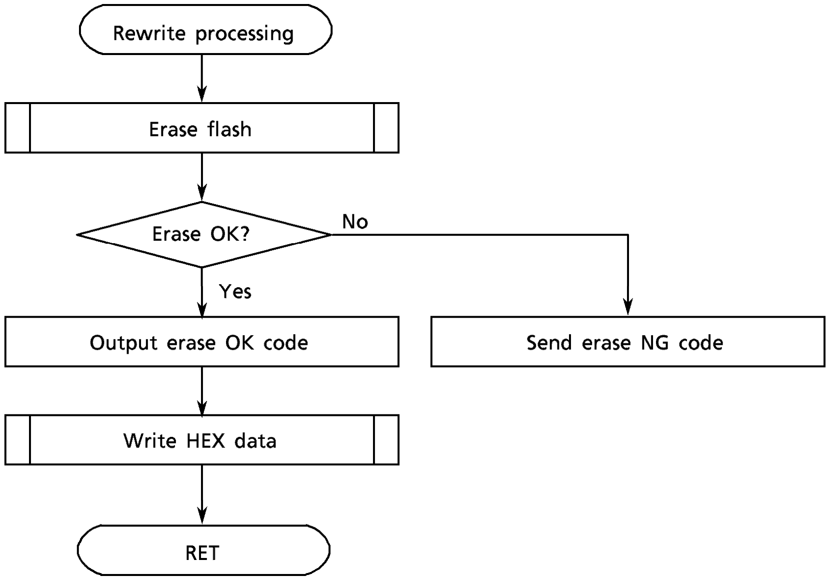
Single Boot General Flow



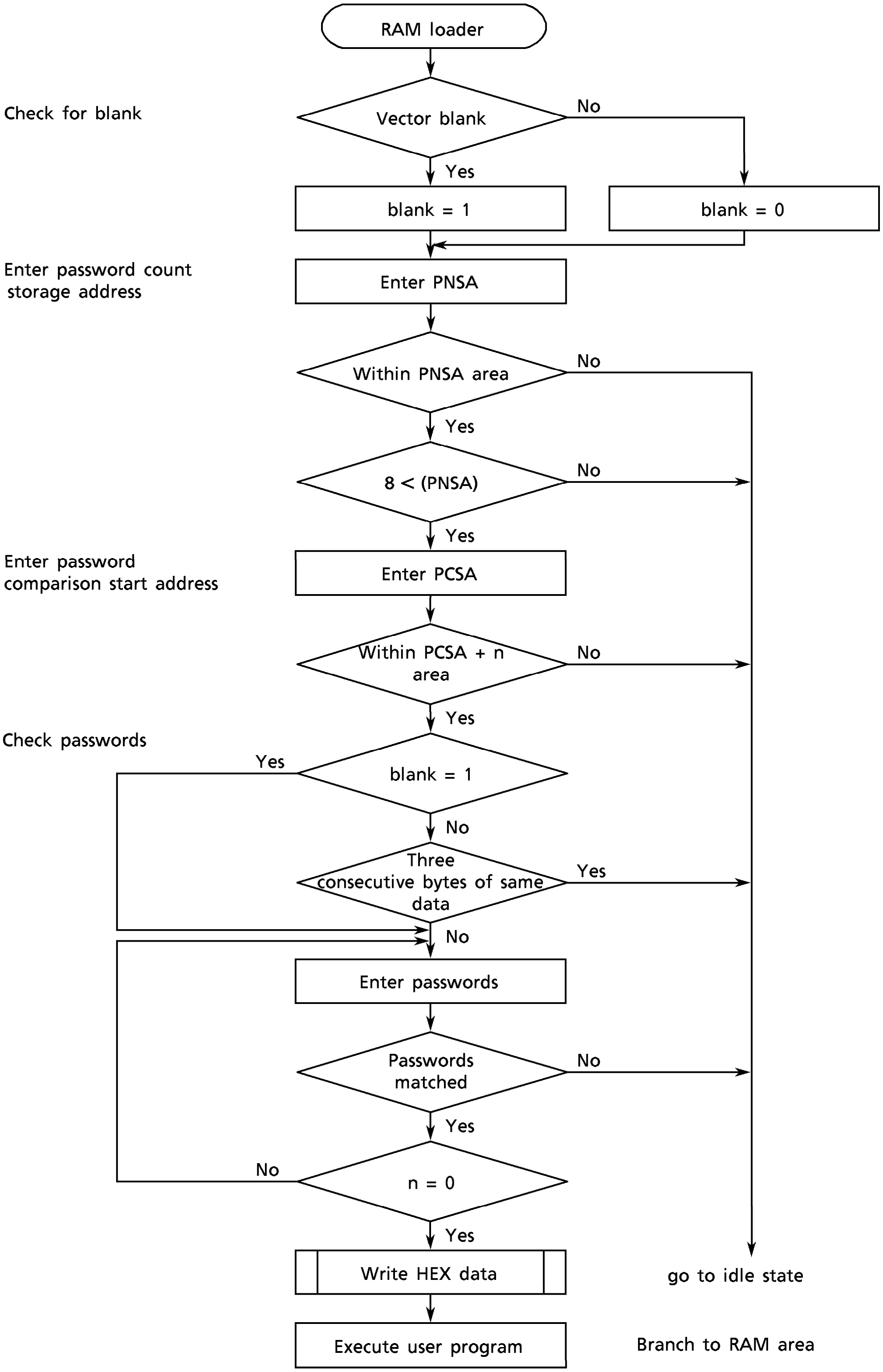
(1) SUM command



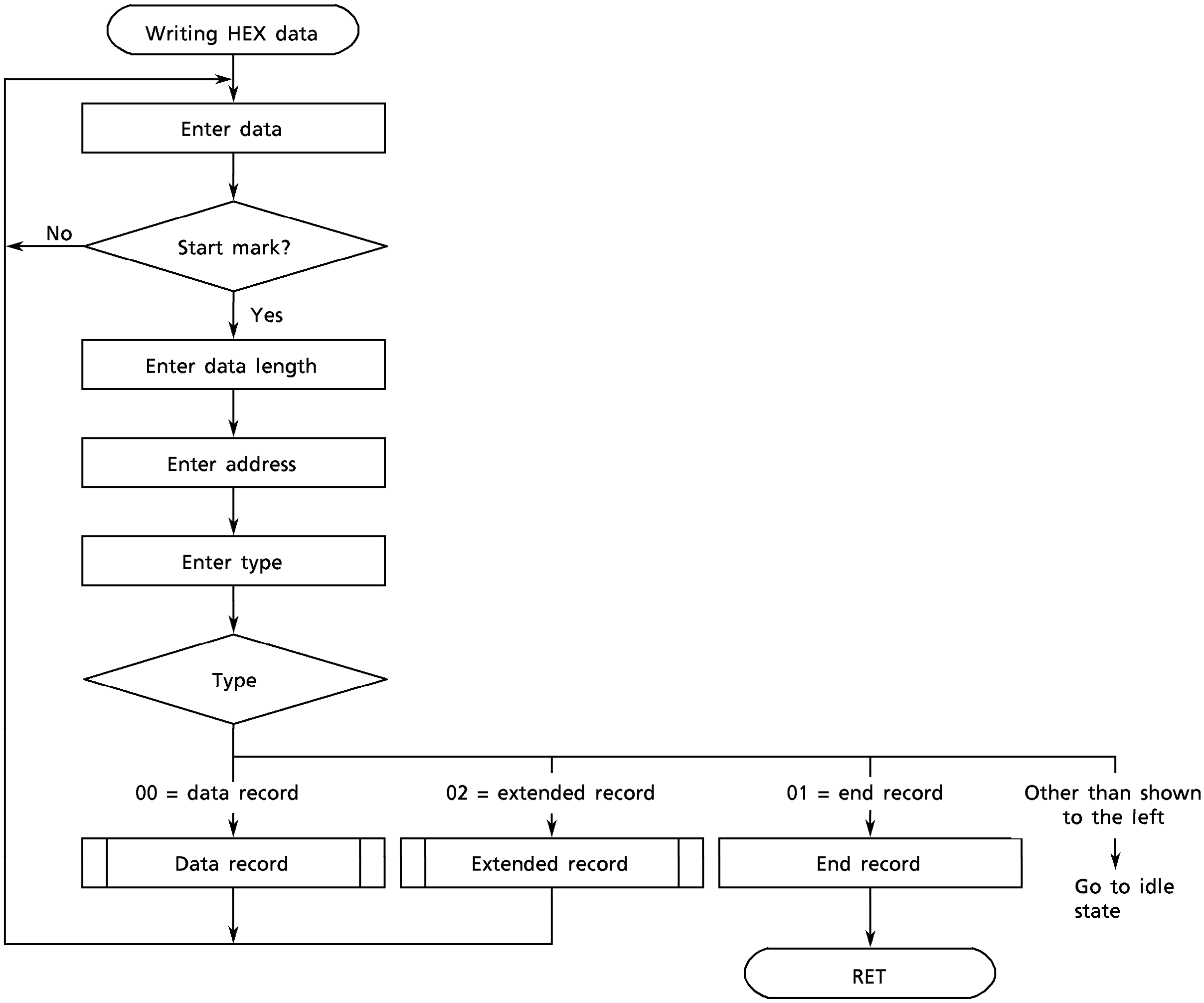
(2) Rewrite command



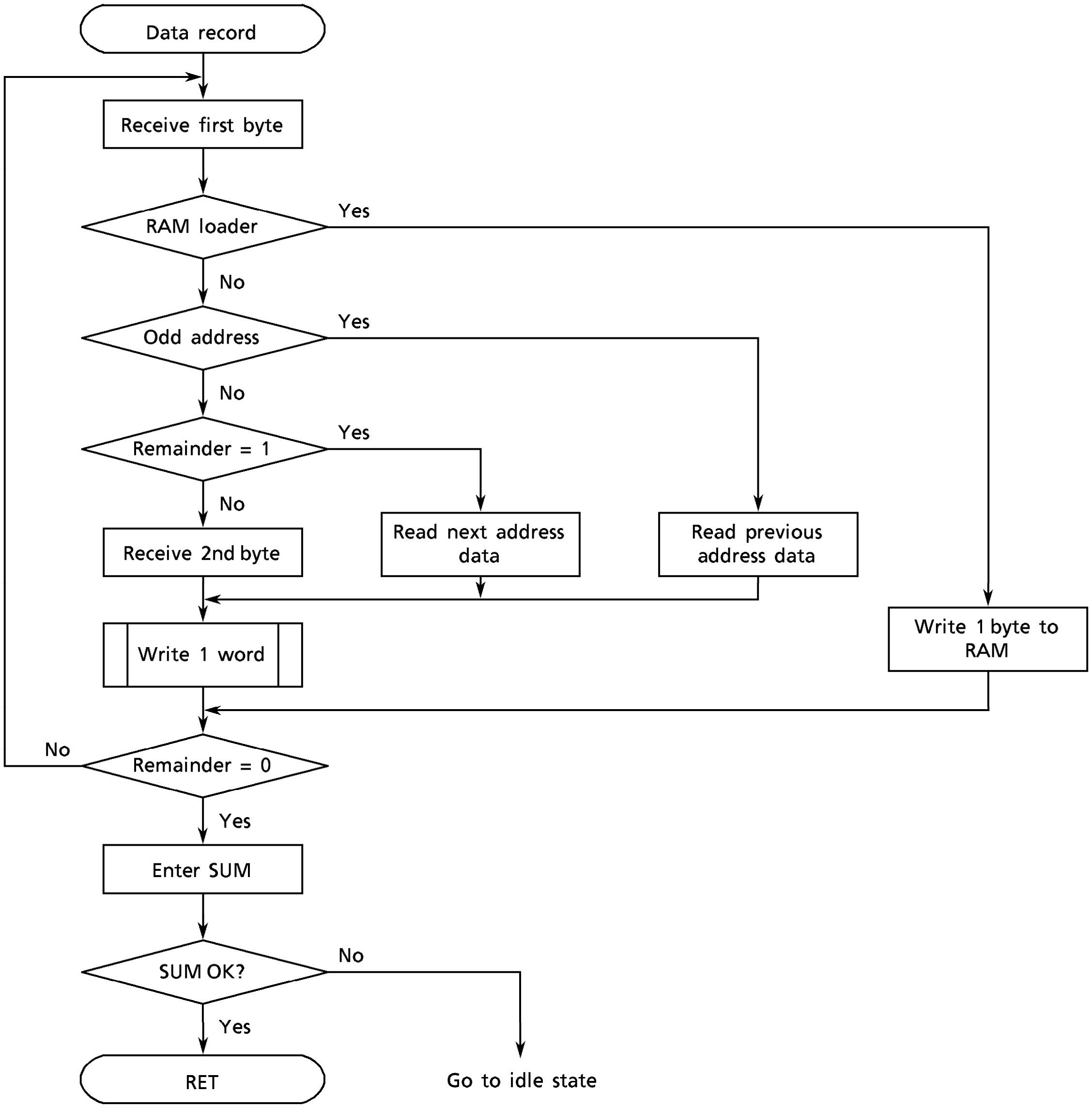
(3) RAM loader command



(2)-1 Writing HEX data

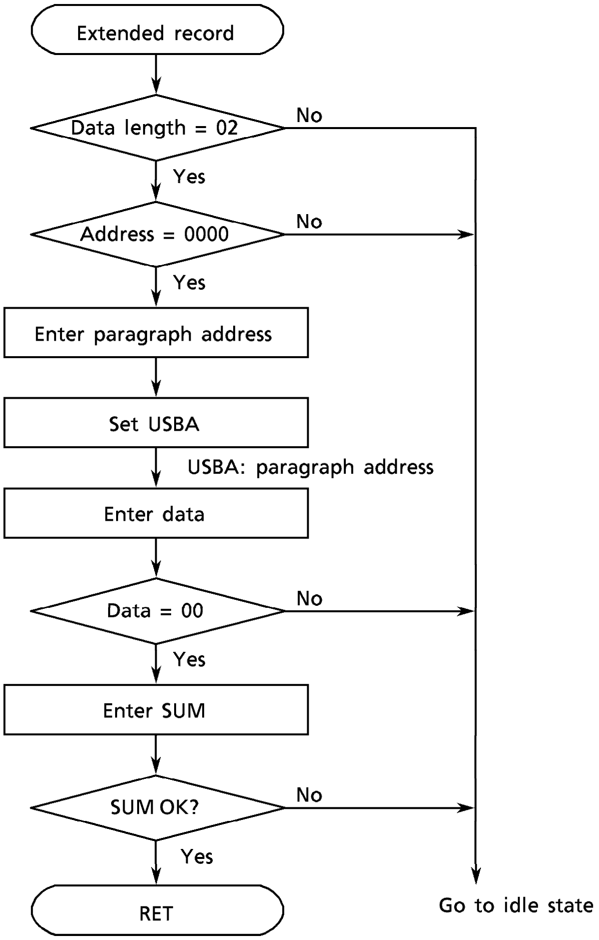


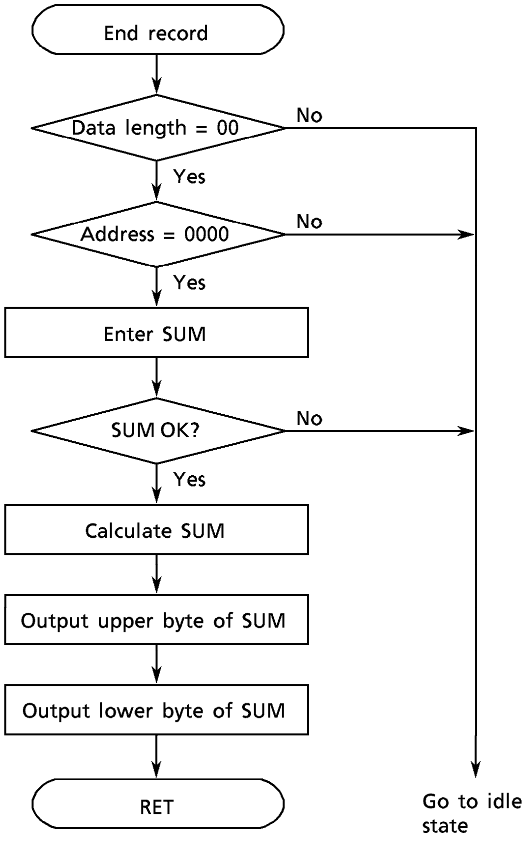
(2)-1-1 Data record



(2)-1-2 Extended record

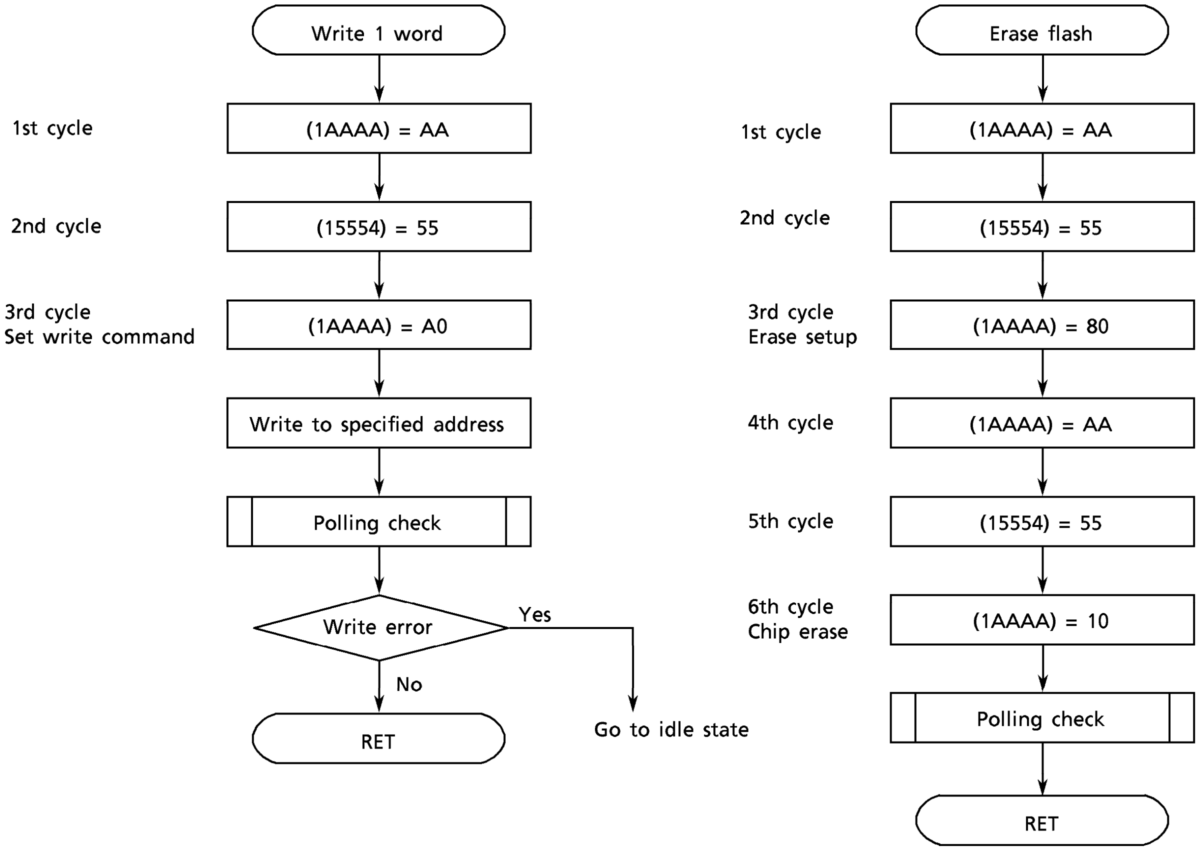
(2)-1-3 End record



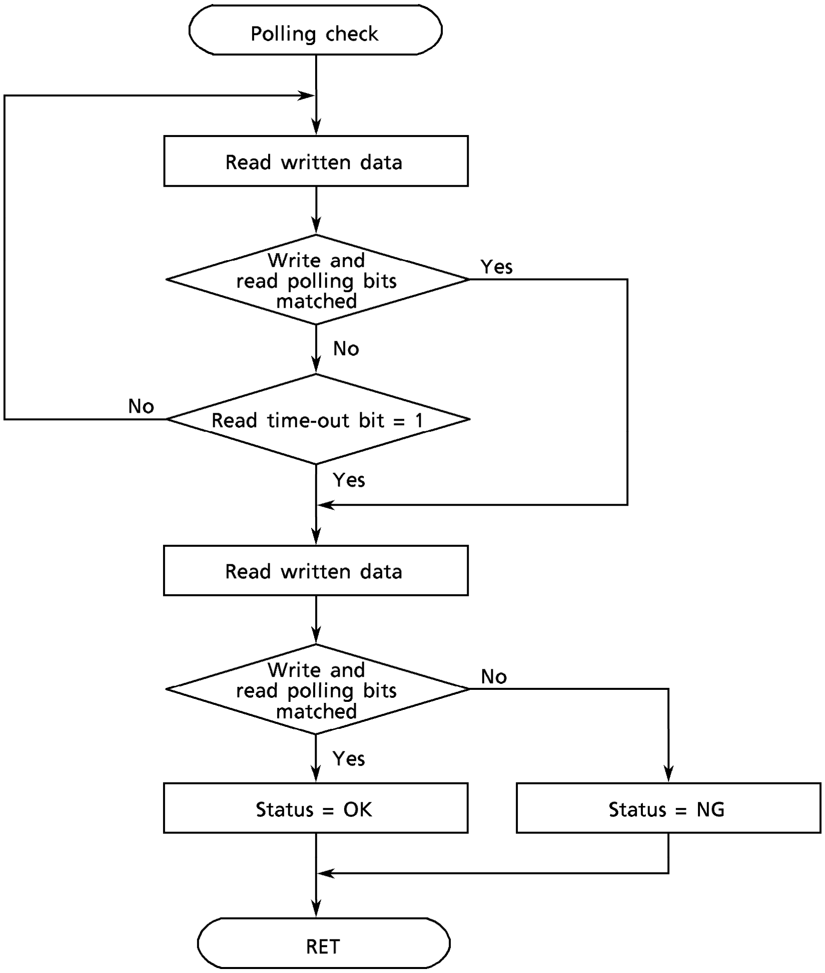


(2)-1-1-1 Writing one word

(2)-2 Flash memory erase



(2)-2-1 Data polling



4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power Supply Voltage	V_{CC}	− 0.5 to + 6.5	V
Input Voltage	V_{IN}	− 0.5 to V_{CC} + 0.5	V
Output Current (Total)	ΣI_{OL}	+ 120	mA
Output Current (Total)	ΣI_{OH}	− 120	mA
Power Dissipation (Ta = + 70°C)	P_D	600	mW
Soldering Temperature (10 s)	T_{SOLDER}	+ 260	°C
Storage Temperature	T_{STG}	− 65 to + 150	°C
Operating Temperature	T_{OPR}	− 20 to + 70	°C
Number of Times Program Erased	N_{EW}	1000	Cycle

Note: The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

4.2 DC Characteristics

(1) V_{CC} = + 5 V ± 10%, Ta = −20 to + 70°C (fc = 8 to 25 MHz)

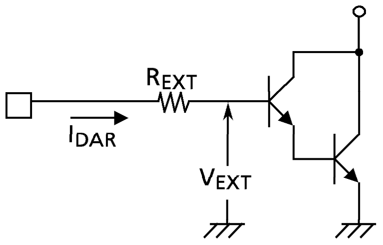
(Single-chip mode, single-boot mode)

Parameter	Symbol	Test Condition	Min	Max	Unit
Low Level Input Voltage (D0 to 15) Port 2 to A (except P56, P70, P72, P73, P75) $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0 to 4 $\overline{\text{EA}}$, AM8/$\overline{16}$ X1	V_{IL} V_{IL1} V_{IL2} V_{IL3} V_{IL4}	—	− 0.3 − 0.3 − 0.3 − 0.3 − 0.3	0.8 0.3 V_{CC} 0.25 V_{CC} 0.3 0.2V_{CC}	V V V V V
High Level Input Voltage (D0 to 15) Port 2 to A (except P56, P70, P72, P73, P75) $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0 to 4 $\overline{\text{EA}}$, AM8/$\overline{16}$ X1	V_{IH} V_{IH1} V_{IH2} V_{IH3} V_{IH4}	—	2.2 0.7 V_{CC} 0.75 V_{CC} V_{CC} − 0.3 0.8V_{CC}	V_{CC} + 0.3 V_{CC} + 0.3 V_{CC} + 0.3 V_{CC} + 0.3 V_{CC} + 0.3	V V V V V
Low Level Output Voltage	V_{OL}	I_{OL} = 1.6 mA		0.45	V
High Level Output Voltage	V_{OH} V_{OH1} V_{OH2}	I_{OH} = − 400 μA I_{OH} = − 100 μA I_{OH} = − 20 μA	2.4 0.75 V_{CC} 0.9 V_{CC}	—	V V V
Output Port Current (8 output pins max.)	I_{DAR}	V_{EXT} = 1.5 V R_{EXT} = 1.1 kΩ	− 1.0	− 3.5	mA
Input Leakage Current Output Leakage Current	I_{LI} I_{LO}	0.0 ≦ Vin ≦ V_{CC} 0.2 ≦ Vin ≦ V_{CC} − 0.2	0.02 (Typ.) 0.05 (Typ.)	± 5 ± 10	μA μA
Operating Current (RUN) IDLE2 IDLE1 STOP (Ta = − 20 to + 70°C)	I_{CC}	fc = 25 MHz 0.2 ≦ Vin ≦ V_{CC} − 0.2		100 40 10 150	mA mA mA μA
Power-down Voltage (@STOP, RAM backup)	V_{STOP}	V_{IL2} = 0.2 V_{CC}, V_{IH2} = 0.8 V_{CC}	2.0	6.0	V
Pull-up Resistance	R_{RP}		45	160	kΩ
Pin Capacitance	C_{IO}	fc = 1 MHz	—	10	pF
Schmitt Width $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0 to 4	V_{TH}		0.4	1.0 (Typ.)	V

Note 1: The Typ. values are referenced to V_{CC} = + 5 V at Ta = + 25°C.

Note 2: The I_{DAR} stipulated above is guaranteed for a total of 8 lines of any output ports used.

Reference diagram: Definition of I_{DAR}



4.3 AC Characteristics

(1) V_{CC} = + 5 V ± 10%, Ta = − 20 to + 70°C

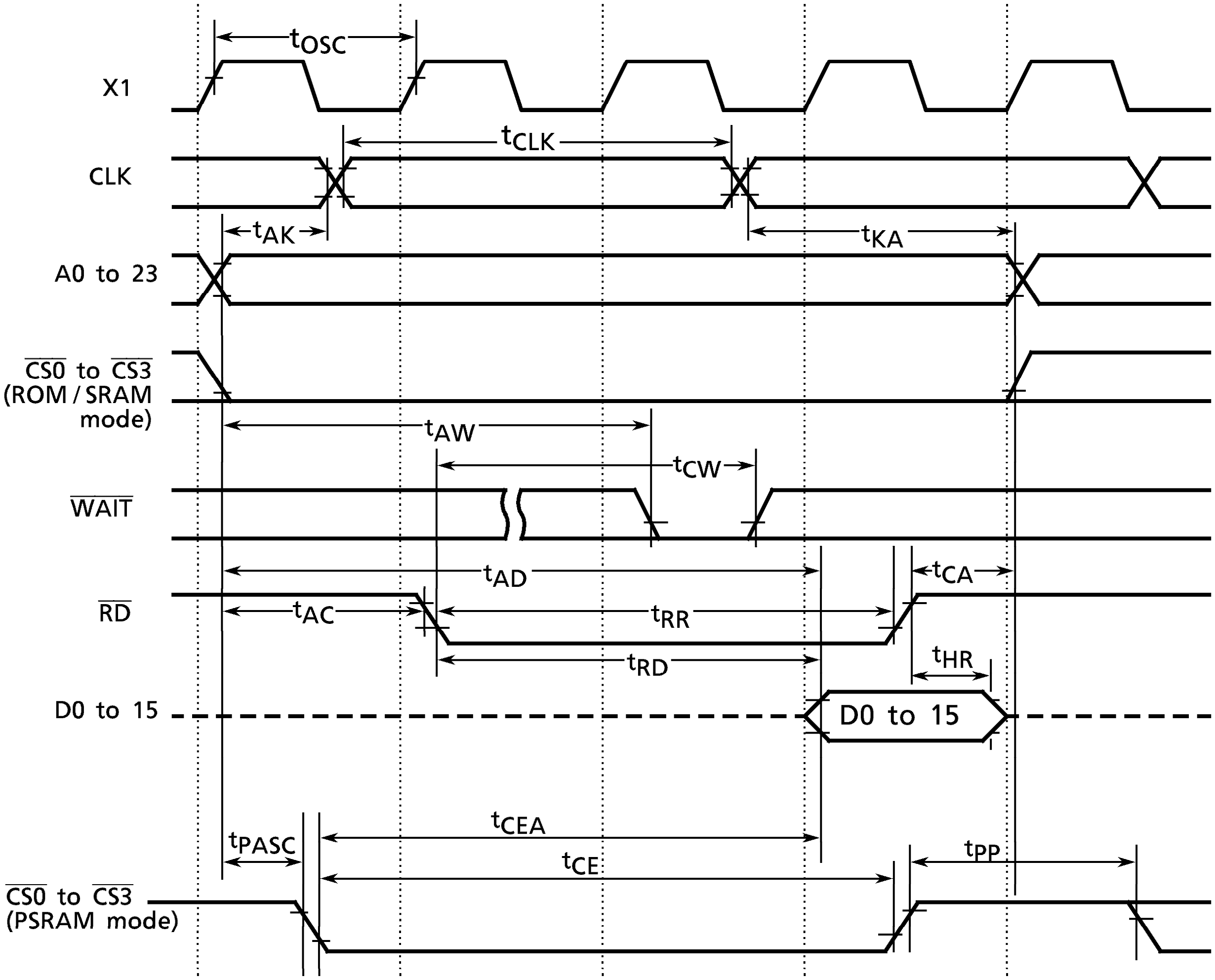
(fc = 8 MHz to 25 MHz)

No.	Parameter	Sym-bol	Variable		20 MHz		25 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Oscillation Period (= x)	t_{OSC}	40	125	50	—	40	—	ns
2	CLK Pulse Width	t_{CLK}	2.0x − 40	22.5	60	—	40	—	ns
3	Valid A0 to 23 → CLK hold	t_{AK}	0.5x − 20	0	5	—	0	—	ns
4	Valid CLK → A0 to 23 hold	t_{KA}	1.5x − 60	0	15	—	0	—	ns
5	Valid A0 to 23 → $\overline{RD}$ / $\overline{WR}$ fall	t_{AC}	1.0x − 20	11.25	30	—	20	—	ns
6	$\overline{RD}$ / $\overline{WR}$ rise → A0 to 23 hold	t_{CA}	0.5x − 20	5	5	—	0	—	ns
7	Valid A0 to 23 → D0 to 15 input	t_{AD}	69.4	3.5x − 40	—	135	—	100	ns
8	$\overline{RD}$ fall → D0 to 15 input	t_{RD}	33.1	2.5x − 45	—	80	—	55	ns
9	$\overline{RD}$ low pulse width	t_{RR}	2.5x − 40	38.1	85	—	60	—	ns
10	$\overline{RD}$ rise → D0 to 15 hold	t_{HR}	0	0	0	—	0	—	ns
11	$\overline{WR}$ low pulse width	t_{WW}	2.5x − 40	38.1	85	—	60	—	ns
12	Valid D0 to 15 → $\overline{WR}$ rise	t_{DW}	2.0x − 40	22.5	60	—	40	—	ns
13	$\overline{WR}$ rise → D0 to 15 hold	t_{WD}	0.5x − 10	5.6	15	—	10	—	ns
14	Valid A0 to 23 → $\overline{WAIT}$ input (1 WAIT + n mode)	t_{AW}	19.4	3.5x − 90	—	85	—	50	ns
	Valid A0 to 23 → $\overline{WAIT}$ input (0 + n WAIT mode)	t_{AW}	6.8	1.5x − 40	—	35	—	20	ns
15	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold (1 WAIT + n mode)	t_{CW}	2.5x + 0	78.1	125	—	100	—	ns
	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold (0 + n WAIT mode)	t_{CW}	0.5x + 0	15.6	25	—	20	—	ns
16	$\overline{WR}$ rise → valid PORT	t_{CP}	—	200	—	200	—	200	ns
17	$\overline{CS}$ low pulse width (PSRAM mode)	t_{CE}	3.0x − 40	—	110	—	80	—	ns
18	$\overline{CS}$ fall → D0 to 15 input (SRAM mode)	t_{CEA}	—	3.0x − 60	—	90	—	60	ns
19	Address setup time (SRAM mode)	t_{PASC}	0.5x − 15	—	10	—	5	—	ns
20	$\overline{CS}$ precharge time (SRAM mode)	t_{PP}	1.0x − 10	—	40	—	30	—	ns

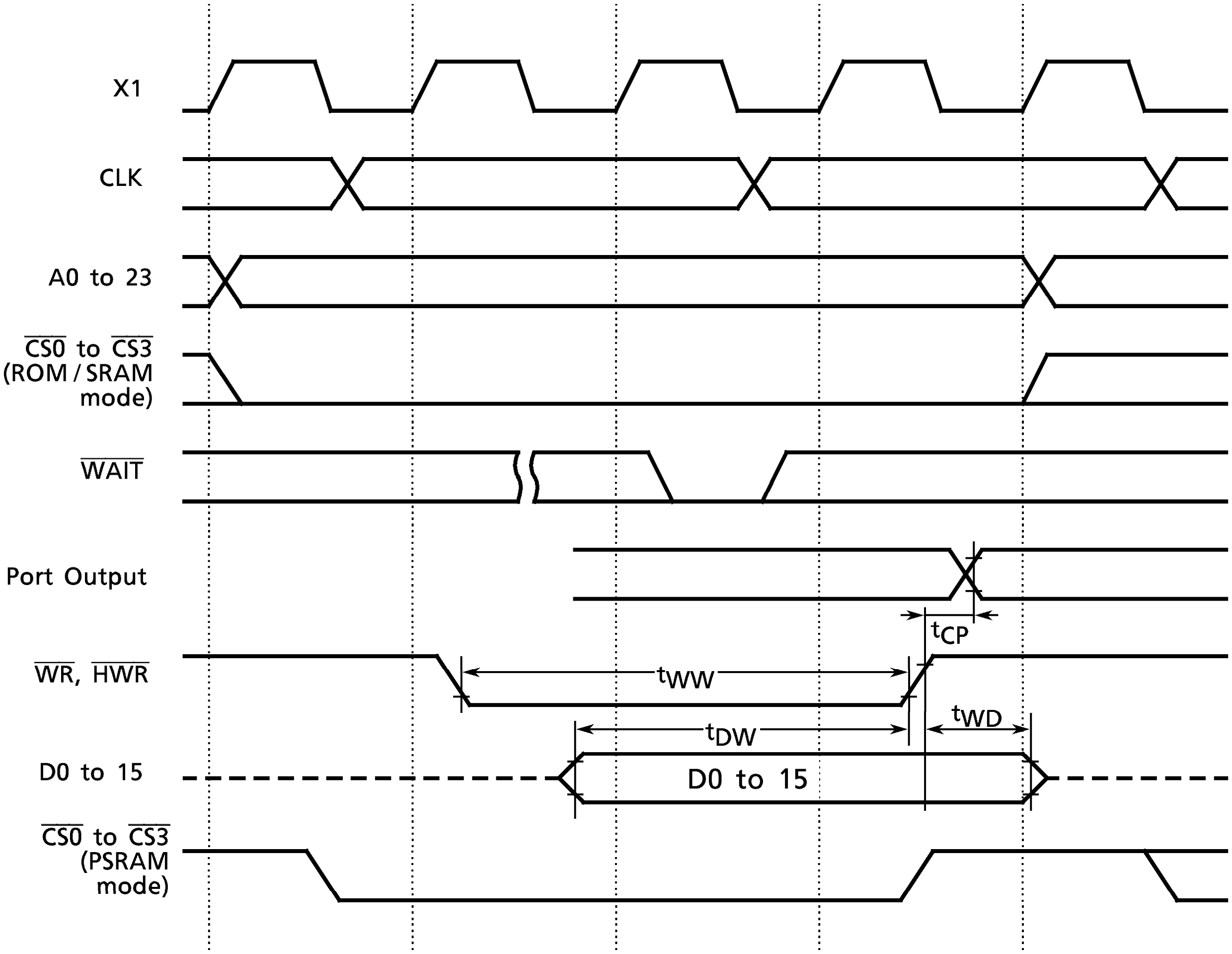
AC test conditions

- Output level: High 2.2 V / Low 0.8 V, CL = 50 pF
- Input level: High 2.4 V / Low 0.45 V (D0 to D15)
  High 0.8 V_{CC} / Low 0.2 Vcc (except for D0 to D15)

(2) Read cycle


t_{OSC}
X1
t_{CLK}
CLK
t_{AK}
t_{KA}
A0 to 23
$\overline{CS0}$ to $\overline{CS3}$ (ROM / SRAM mode)
t_{AW}
t_{CW}
$\overline{WAIT}$
t_{AD}
t_{CA}
$\overline{RD}$
t_{AC}
t_{RR}
t_{RD}
t_{HR}
D0 to 15
D0 to 15
t_{PASC}
t_{CEA}
t_{CE}
t_{PP}
$\overline{CS0}$ to $\overline{CS3}$ (PSRAM mode)

(3) Write cycle



4.4 Serial Channel Timing

(1) I/O interface modes

① SCLK input mode

V_{CC} = + 5 V ± 10%, Ta = – 20 to + 70°C (fc = 8 to 25 MHz)

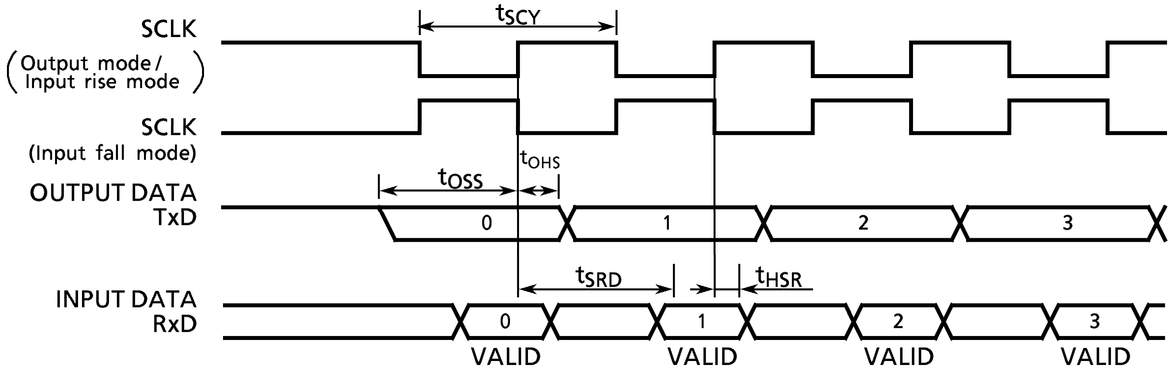
Parameter	Symbol	Variable		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK period	t_{SCY}	16x	—	1.6	—	0.64	—	μs
Output data → SCLK rise/fall*	t_{OSS}	t_{SCY}/2 – 5x – 50	—	250	—	70	—	ns
SCLK rise/fall* → Output data hold	t_{OHS}	5x – 100	—	400	—	100	—	ns
SCLK rise/fall* → Input data hold	t_{HSR}	0	—	0	—	0	—	ns
SCLK rise/fall* → Valid data input	t_{SRD}	—	t_{SCY} – 5x – 100	—	1000	—	340	ns

*) SCLK rise/fall refers to the timing at which SCLK rises when in SCLK rising edge mode or at which SCLK falls when in SCLK falling edge mode.

② SCLK output mode

V_{CC} = + 5 V ± 10%, Ta = – 20 to + 70°C (fc = 8 to 25 MHz)

Parameter	Symbol	Variable		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK period (programmable)	t_{SCY}	16x	8192x	1.6	819.2	0.64	327.6	μs
Output data → SCLK rise	t_{OSS}	t_{SCY} – 2x – 150	—	1250	—	410	—	ns
SCLK rise → Output data hold	t_{OHS}	2x – 80	—	120	—	0	—	ns
SCLK rise → Input data hold	t_{HSR}	0	—	0	—	0	—	ns
SCLK rise → Valid data input	t_{SRD}	—	t_{SCY} – 2x – 150	—	1250	—	410	ns



(2) UART mode (SCLK0 to 2 external inputs)

V_{CC} = + 5 V ± 10%, Ta = − 20 to + 70°C (fc = 8 to 25 MHz)

Parameter	Symbol	Variable		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Period	t_{SCY}	4x + 20	—	420	—	180	—	ns
SCLK Low Level Pulse Width	t_{SCYL}	2x + 5	—	205	—	85	—	ns
SCLK High Level Pulse Width	t_{SCYH}	2x + 5	—	205	—	85	—	ns

4.5 A / D Conversion Characteristics

V_{CC} = + 5 V ± 10%, Ta = − 20 to + 70°C (fc = 8 to 25 MHz)

Parameter		Sym-bol	Min	Typ.	Max	Unit
A / D Analog Reference Supply Voltage (+)		V_{REFH}	V_{CC} − 0.2	—	V_{CC}	V
A / D Analog Reference Supply Voltage (−)		V_{REFL}	V_{SS}	—	V_{SS} + 0.2	
Analog Reference Voltage		AV_{CC}	V_{CC} − 0.2	—	V_{CC}	
Analog Reference Voltage		AV_{SS}	V_{SS}	—	V_{SS} + 0.2	
Analog Input Voltage		V_{AIN}	V_{REFL}	—	V_{REFH}	
Analog Reference Voltage Power Supply Current	<VREFON> = 1	I_{REF}	—	—	3.7	mA
	<VREFON> = 0		—	0.02	5.0	μA
Overall Error (Not Including Quantization Error)		E_T	—	± 1	± 3	LSB

Note 1: 1LSB = (VREFH − VREFL) / 2^{10} [V]

Note 2: The power supply current flowing from the AV_{CC} pin is included in the power supply current I_{CC} of the V_{CC} pin.

4.6 D / A Conversion Characteristics

V_{CC} = + 5 V ± 10%, Ta = − 20 to + 70°C (fc = 8 to 25 MHz)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog Reference Voltage	AV_{CC}	—	V_{CC} − 0.2	—	V_{CC}	V
Analog Reference Voltage	AV_{SS}		V_{SS}	—	V_{SS} + 0.2	
Overall Error	—	R = 1 MΩ (Note) R = 5 MΩ (Note) R = 10 MΩ (Note)	—	—	7.0 4.0 3.5	LSB LSB LSB
Differential Linearity Error		—	—	2.0	—	LSB

Note: R denotes the external load resistance of D / A converter output pins (DAOUT0, DAOUT1).

4.7 Event Counter (External Input Clocks: TI0, TI4, TI8, TI9, TIA, TIB)

V_{CC} = + 5 V ± 10%, Ta = − 20 to + 70°C (fc = 8 to 25 MHz)

Parameter	Symbol	Variable		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
External Input Clock Period	t_{VCK}	8x + 100	—	900	—	420	—	ns
External Input Clock Low Level Pulse Width	t_{VCKL}	4x + 40	—	440	—	200	—	ns
External Input Clock High Level Pulse Width	t_{VCKH}	4x + 40	—	440	—	200	—	ns

4.8 Interrupt Operation

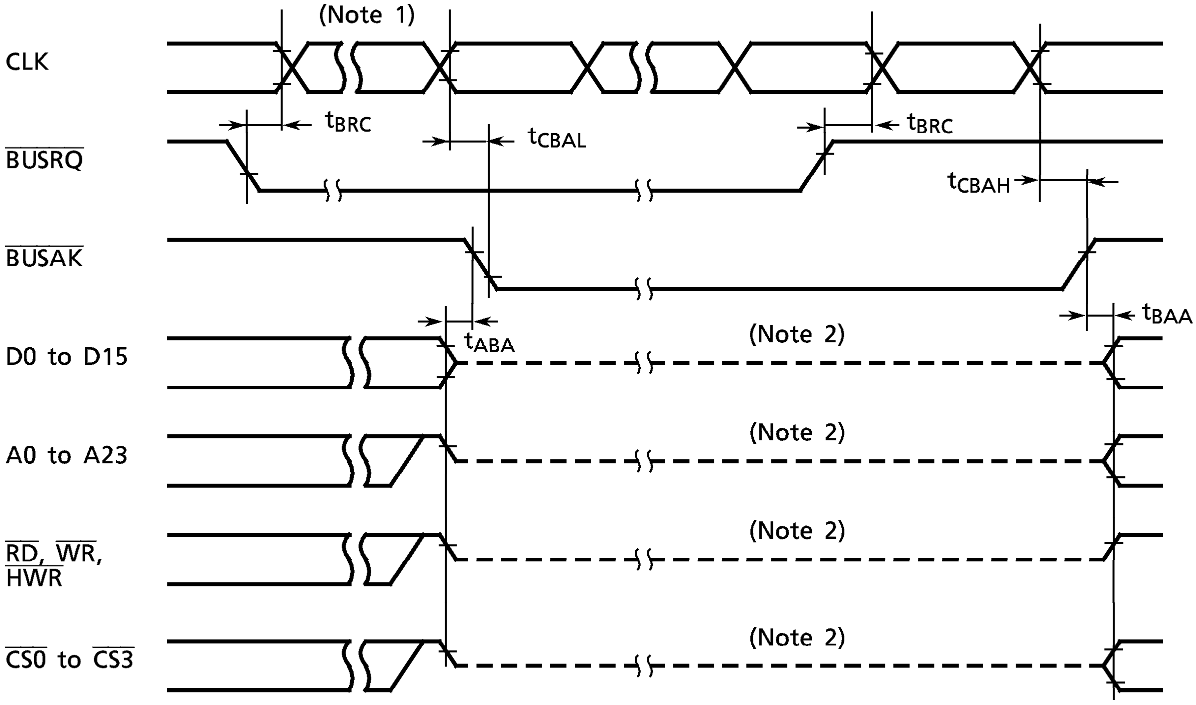
V_{CC} = + 5 V ± 10%, Ta = − 20 to + 70°C (fc = 8 to 25 MHz)

Parameter	Symbol	Variable		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{\text{NMI}}$ and INT0 to 4 Low Level Pulse Width	t_{INTAL}	4x	—	400	—	160	—	ns
$\overline{\text{NMI}}$ and INT0 to 4 High Level Pulse Width	t_{INTAH}	4x	—	400	—	160	—	ns
INT5 to INT8 Low Level Pulse Width	t_{INTBL}	8x + 100	—	900	—	420	—	ns
INT5 to INT8 High Level Pulse Width	t_{INTBH}	8x + 100	—	900	—	420	—	ns

4.9 Bus Request / Bus Acknowledge Timing

V_{CC} = + 5 V ± 10%, Ta = − 20 to + 70°C (fc = 8 to 25 MHz)

Parameter	Symbol	Variable		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{\text{BUSRQ}}$ setup time relative to CLK	t_{BRC}	120	—	120	—	120	—	ns
CLK→$\overline{\text{BUSAK}}$ fall	t_{CBAL}	—	2.0x + 120	—	320	—	200	ns
CLK→$\overline{\text{BUSAK}}$ rise	t_{CBAH}	—	0.5x + 40	—	90	—	60	ns
Duration from output buffer turn-off to $\overline{\text{BUSAK}}$ fall	t_{ABA}	0	80	0	80	0	80	ns
Duration from $\overline{\text{BUSAK}}$ rise to output buffer turn-on	t_{BAA}	0	80	0	80	0	80	ns



Note 1: If when control of the bus is requested by pulling $\overline{\text{BUSRQ}}$ low, the preceding bus cycle is not completed due to wait state, the bus will not be relinquished until exiting from the wait state.

Note 2: The broken line only indicates that the output buffer is turned off, and not that the signal is at an intermediate voltage level. Immediately after the bus is relinquished, the immediately preceding signal level is dynamically held on by an external load capacitance. Therefore, if the signal level needs to be set high or low using an external resistor, the signal level setup immediately after relinquishing the bus may be delayed by the external load capacitance (RC time constant). Consider this delay when designing your system. The built-in programmable pullup resistors continue working depending on the internal signal state.