TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/L1 Series

TMP91FY12A

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 to 4, INTRTC) 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 5 clocks of f_{FPH}) 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 Microcontrollers

TMP91FY12AF

1. Outline and Features

TMP91FY12AF is a high-speed 16-bit microcontroller designed for the control of various mid- to large-scale equipment.

TMP91FY12AF comes in a 100-pin flat package.

Listed below are the features.

(1) High-speed 16-bit CPU (900/L1 CPU)
- Instruction mnemonics are upward-compatible with TLCS-90/900
- 16 Mbytes of linear address space
- General-purpose registers and register banks
- 16-bit multiplication and division instructions; bit transfer and arithmetic instructions
- Micro DMA: 4 channels (1.0 μs/2 bytes at 16 MHz)

(2) Minimum instruction execution time: 148 ns (at 27 MHz)

(3) Built-in RAM: 4 Kbytes
Built-in ROM: 256 Kbytes Flash memory
2 Kbytes mask ROM (used for booting)

(4) External memory expansion
- Expandable up to 16 Mbytes (shared program/data area)
- Can simultaneously support 8-/16-bit width external data bus
  ・・・Dynamic data bus sizing

(5) 8-bit timers: 8 channels

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

(7) General-purpose serial interface: 2 channels
- UART/Synchronous mode: 2 channels
- IrDA ver 1.0 (115.2 kbps) supported: 1 channel

030519EBP1

- The information contained herein is subject to change without notice.
- 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.
- 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 malfunctionor 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.
- TOSHIBA products should not be embedded to the downstream products which are prohibited to be produced and sold, under any law and regulations.
- 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.

Purchase of TOSHIBA I²C components conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

(8) Serial bus interface: 1 channel
- I^2C bus mode/clock synchronous select mode

(9) 10-bit AD converter (sample-hold circuit is built in): 8 channels

(10) Watchdog timer

(11) Timer for real-time clock (RTC)

(12) Chip Select/Wait controller: 4 channels

(13) Interrupts: 45 interrupts
- 9 CPU interrupts: Software interrupt instruction and illegal instruction
- 26 internal interrupts: 7-level priority can be set.
- 10 external interrupts: 7-level priority can be set.

(14) Input/output ports: 81 pins

(15) Standby function
Three Halt modes: Idle2 (programmable), Idle1, Stop

(16) Triple-clock controller
- Clock Doubler (DFM)
- Clock Gear (fc to fc/16)
- Slow mode (fs = 32.768 kHz)

(17) Operating voltage
- V_{CC} = 2.7 V to 3.6 V (fc max = 27 MHz)

(18) Package
- 100-pin QFP: P-QFP100-1414-0.50E

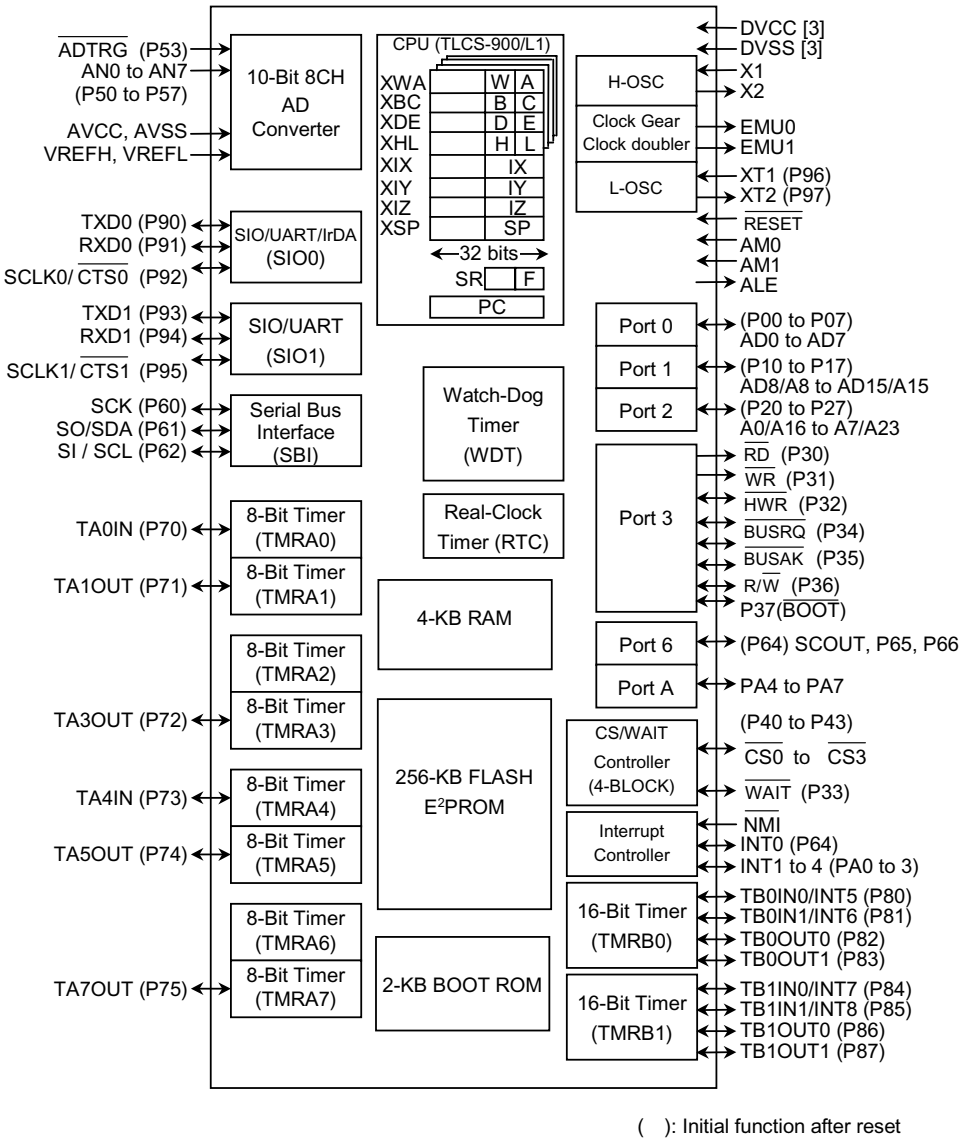


Figure 1.1 TMP91FY12AF Block Diagram

2. Pin Assignment and Pin Functions

The assignment of input/output pins for the TMP91FY12AF, their names and functions are as follows:

2.1 Pin Assignment Diagram

Figure 2.1.1 shows the pin assignment of the TMP91FY12AF.

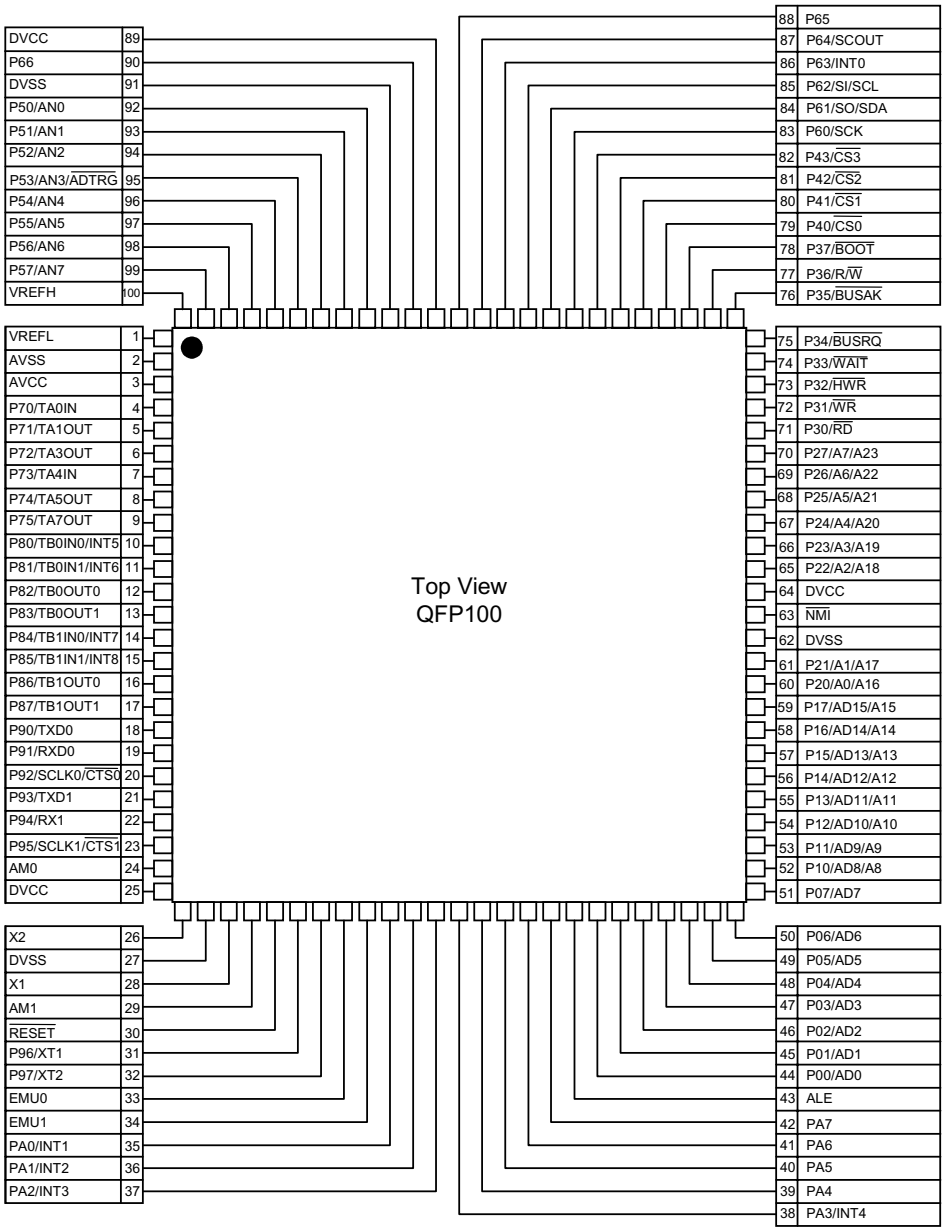


Figure 2.1.1 Pin assignment diagram (100-pin QFP)

2.2 Pin Names and Functions

The names of the input/output pins and their functions are described below.
Table 2.2.1 Pin names and functions.

Table 2.2.1 Pin names and functions (1/3)

Pin Name	Number of Pins	I/O	Functions
P00 to P07	8	I/O	Port 0: I/O port that allows I/O to be selected at the bit level
AD0 to AD7		Tri-state	Address and data (lower): Bits 0 to 7 of address and data bus
P10 to P17	8	I/O	Port 1: I/O port that allows I/O to be selected at the bit level
AD8 to AD15		Tri-state	Address and data (upper): Bits 8 to 15 for address and data bus
A8 to A15		Output	Address: Bits 8 to 15 of address bus
P20 to P27	8	I/O	Port 2: I/O port that allows I/O to be selected at the bit level (with pull-down resistor)
A0 to A7		Output	Address: Bits 0 to 7 of address bus
A16 to A23		Output	Address: Bits 16 to 23 of address bus
P30	1	Output	Port 30: Output port
$\overline{RD}$		Output	Read: Strobe signal for reading external memory
P31	1	Output	Port 31: Output port
$\overline{WR}$		Output	Write: Strobe signal for writing data to pins AD0 to AD7
P32	1	I/O	Port 32: I/O port (with pull-up resistor)
$\overline{HWR}$		Output	High Write: Strobe signal for writing data to pins AD8 to AD15
P33	1	I/O	Port 33: I/O port (with pull-up resistor)
$\overline{WAIT}$		Input	Wait: Pin used to request CPU bus wait
P34	1	I/O	Port 34: I/O port (with pull-up resistor)
$\overline{BUSRQ}$		Input	Bus Request: Signal used to request Bus Release
P35	1	I/O	Port 35: I/O port (with pull-up resistor)
$\overline{BUSAK}$		Output	Bus Acknowledge: Signal used to acknowledge Bus Release
P36	1	I/O	Port 36: I/O port (with pull-up resistor)
R/ $\overline{W}$		Output	Read/Write: 1 represents Read or Dummy cycle; 0 represents Write cycle.
P37	1	I/O	Port 37: I/O port (with pull-up resistor)
$\overline{BOOT}$		Input	This pin sets single boot mode.
P40	1	I/O	Port 40: I/O port (with pull-up resistor)
$\overline{CS0}$		Output	Chip Select 0: Outputs 0 when address is within specified address area
P41	1	I/O	Port 41: I/O port (with pull-up resistor)
$\overline{CS1}$		Output	Chip Select 1: Outputs 0 if address is within specified address area
P42	1	I/O	Port 42: I/O port (with pull-up resistor)
$\overline{CS2}$		Output	Chip Select 2: Outputs 0 if address is within specified address area
P43	1	I/O	Port 43: I/O port (with pull-up resistor)
$\overline{CS3}$		Output	Chip Select 3: Outputs 0 if address is within specified address area
P50 to P57	8	Input	Port 5: pin used to input port
AN0 to AN7		Input	Analog input: Pin used to input to AD converter
$\overline{ADTRG}$		Input	AD Trigger: Signal used to request start of AD converter
P60	1	I/O	Port 60: I/O port
SCK		I/O	Serial bus interface clock in SIO Mode
P61	1	I/O	Port 61: I/O port
SO		Output	Serial bus interface output data in SIO Mode
SDA		I/O	Serial bus interface data in I^2C bus Mode
P62	1	I/O	Port 62: I/O port
SI		Input	Serial bus interface input data in SIO Mode
SCL		I/O	Serial bus interface clock in I^2C bus Mode
P63	1	I/O	Port 63: I/O port
INT0		Input	Interrupt Request Pin 0: Interrupt request pin with programmable level / rising edge / falling edge
P64	1	I/O	Port 64: I/O port
SCOUT		Output	System Clock Output: Outputs f_{FPH} or fs clock.

Table 2.2.1 Pin names and functions (2/3)

Pin Name	Number of Pins	I/O	Functions
P65	1	I/O	Port 65: I/O port
P66	1	I/O	Port 66: I/O port
P70 TA0IN	1	I/O Input	Port 70: I/O port Timer A0 Input
P71 TA1OUT	1	I/O Output	Port 71: I/O port Timer A1 Output
P72 TA3OUT	1	I/O Output	Port 72: I/O port Timer A3 Output
P73 TA4IN	1	I/O Input	Port 73: I/O port Timer A4 Input
P74 TA5OUT	1	I/O Output	Port 74: I/O port Timer A5 Output
P75 TA7OUT	1	I/O Output	Port 75: I/O port Timer A7 Output
P80 TB0IN0 INT5	1	I/O Input Input	Port 80: I/O port Timer B0 Input 0 Interrupt Request Pin 5: Interrupt request pin with programmable rising edge/falling edge.
P81 TB0IN1 INT6	1	I/O Input Input	Port 81: I/O port Timer B0 Input 1 Interrupt Request Pin 6: Interrupt request on rising edge
P82 TB0OUT0	1	I/O Output	Port 82: I/O port Timer B0 Output 0
P83 TB0OUT1	1	I/O Output	Port 83: I/O port Timer B0 Output 1
P84 TB1IN0 INT7	1	I/O Input Input	Port 84: I/O port Timer B1 Input 0 Interrupt Request Pin 7: Interrupt request pin with programmable rising edge/falling edge.
P85 TB1IN1 INT8	1	I/O Input Input	Port 85: I/O port Timer B1 Input 1 Interrupt Request Pin 8: Interrupt request on rising edge
P86 TB1OUT0	1	I/O Output	Port 86: I/O port Timer B1 Output 0
P87 TB1OUT1	1	I/O Output	Port 87: I/O port Timer B1 Output 1
P90 TXD0	1	I/O Output	Port 90: I/O port Serial Send Data 0 (Programmable open-drain)
P91 RXD0	1	I/O Input	Port 91: I/O port Serial Receive Data 0
P92 SCLK0 $\overline{CTS0}$	1	I/O I/O Input	Port 92: I/O port Serial Clock I/O 0 Serial Data Send Enable 0 (Clear to Send)
P93 TXD1	1	I/O Output	Port 93: I/O port Serial Send Data 1 (Programmable open-drain)
P94 RXD1	1	I/O Input	Port 94: I/O port (with pull-up resistor) Serial Receive Data 1
P95 SCLK1 $\overline{CTS1}$	1	I/O I/O Input	Port 95: I/O port (with pull-up resistor) Serial Clock I/O 1 Serial Data Send Enable 1 (Clear to Send)
P96 XT1	1	I/O Input	Port 96: I/O port (Open-drain output) Low-frequency oscillator connection pin

Table 2.2.1 Pin names and functions (3/3)

Pin Name	Number of Pins	I/O	Functions
P97 XT2	1	I/O Output	Port 97: I/O port (Open-drain output) Low-frequency oscillator connection pin
PA0 to PA3 INT1 to INT4	4	I/O Input	Ports A0 to A3: I/O ports Interrupt Request Pins 1 to 4: Interrupt request pins with programmable rising edge/falling edge.
PA4 to PA7	4	I/O	Ports A4 to A7: I/O ports
ALE	1	Output	Address Latch Enable Can be disabled to reduce noise.
$\overline{\text{NMI}}$	1	Input	Non-Maskable Interrupt Request Pin: Interrupt request pin with programmable falling edge or both edge.
AM0 to 1	2	Input	Address Mode: The Vcc pin should be connected.
EMU0/EMU1	1	Output	Test Pins: Open pins
$\overline{\text{RESET}}$	1	Input	Reset: initializes TMP91FY12A. (With pull-up resistor)
VREFH	1	Input	Pin for reference voltage input to AD converter (H)
VREFL	1	Input	Pin for reference voltage input to AD converter (L)
AVCC	1	I/O	High-frequency oscillator connection pins
AVSS	1		Power supply pin for AD converter
X1/X2	2		GND pin for AD converter (0 V)
DVCC	3		Power supply pins (All VCC pins should be connected with the power supply pin.)
DVSS	3		GND pins (0 V) (All VSS pins should be connected with the power supply pin.)

Note: An external DMA controller cannot access the device's built-in memory or built-in I/O devices using the $\overline{\text{BUSRQ}}$ and $\overline{\text{BUSAK}}$ signal.

3. Functional Description

This section shows the hardware configuration of the TMP91FY12A 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. The configuration and the functionality of this device are the same as those of the TMP91CW12A. For the functions of this device that are not described here, refer to the TMP91CW12A 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 (including when the watchdog timer output is connected to reset (inside the chip)).

- 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			
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (P37)	AM0	AM1
Single-chip mode	↑	H	H	H
Single-boot mode		L		

3.2 Memory Map

The memory map of this device differs from that of the TMP91CW12A.

Figure 3.2.1 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.

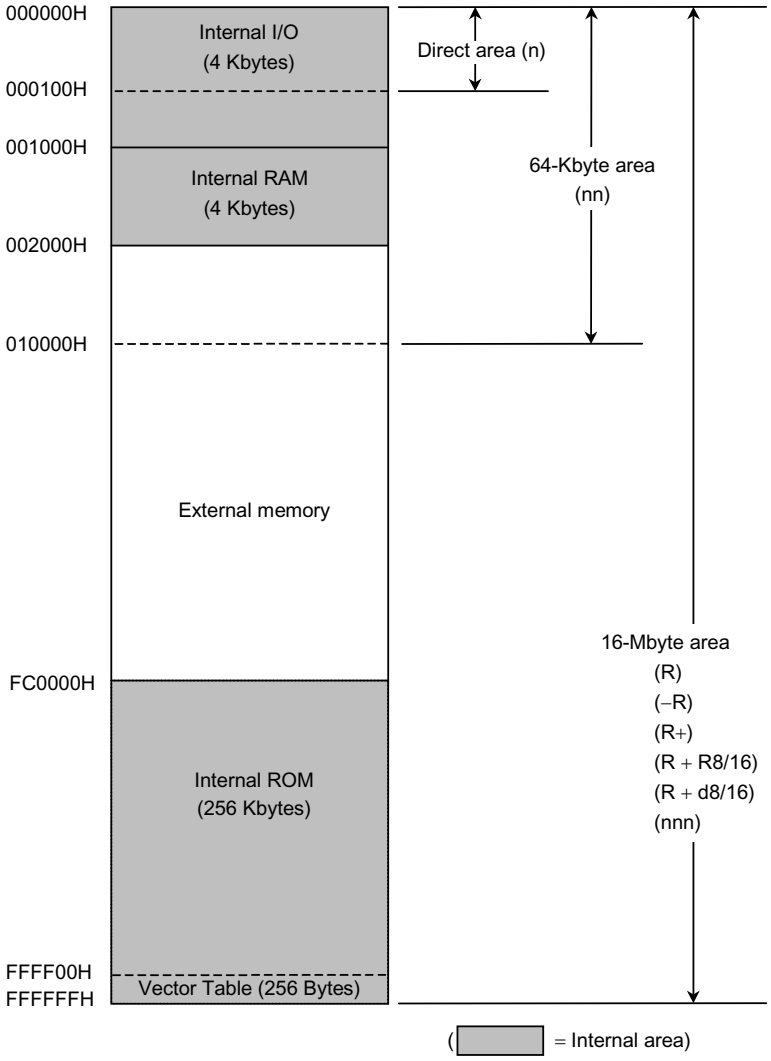


Figure 3.2.1 Memory Map (Single-chip Mode)

3.3 Flash memory

The TMP91FY12A contains an electrically erasable and programmable flash memory using a single 3 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, of multiple blocks together.

Features:

- Program/Erase power supply voltage

  Vcc = 2.7 to 3.6 V

- Structure

  256 K x 8-bit/
  128 K x 16-bit (256 Kbyte)

- Functions

  Automatic program
  Automatic erase
  Automatic multiblock erase
  Data Polling/toggle bit

- Block erase architecture

  16 Kbytes x 1/8 Kbytes x 2/
  32 Kbytes x 1/64 Kbytes x 3

- Mode control

  Based on standard JEDEC commands

- General-purpose flash memory type

  Equivalent to 29LV400T

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

Block structure:

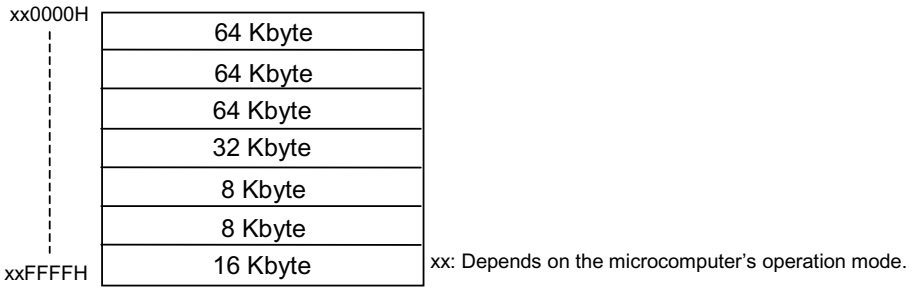


Figure 3.3.1 Block structure of the Flash Memory

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

Command Sequence	Bus cycle	1st bus write cycle		2nd bus write cycle		3rd bus write cycle		4th bus read/write cycle		5th bus write cycle		6th 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	CPUaddress: 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	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
xAAAAH	memory	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
x5554H	address area	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. Write 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(on erasing 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, 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	x	x	x	FC0000H to FCFFFFH	010000H to 01FFFFH	64 Kbyte
BA1	L	H	x	x	x	FD0000H to FDFFFFH	020000H to 02FFFFH	64 Kbyte
BA2	H	L	x	x	x	FE0000H to FEFFFFH	030000H to 03FFFFH	64 Kbyte
BA3	H	H	L	x	x	FF0000H to FF7FFFH	040000H to 047FFFH	32 Kbyte
BA4	H	H	H	L	L	FF8000H to FF9FFFH	048000H to 049FFFH	8 Kbyte
BA5	H	H	H	L	H	FFA000H to FFBFFFH	04A000H to 04BFFFH	8 Kbyte
BA6	H	H	H	H	x	FFC000H to FFFFFFH	04C000H to 04FFFFH	16 Kbyte

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 users 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 date 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 ease 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 address of 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 e 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 the 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 complete, 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 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. Therefor, 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) RDY/BSY (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 micorocomputer 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."

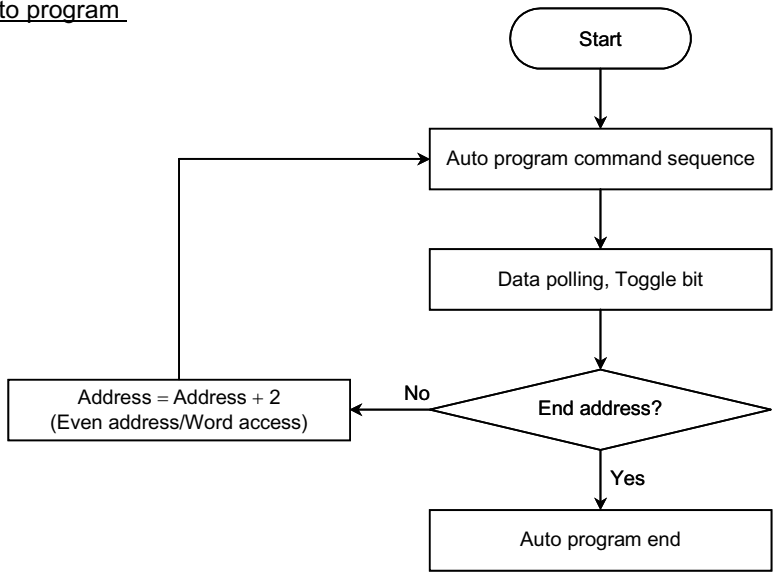
2) User boot:

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.

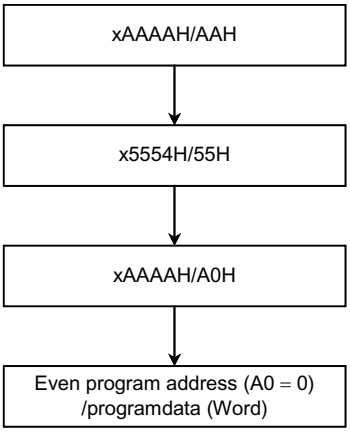
The flash memory rewrite program including routines for taking in the rewrite data and rewriting the flash memory needs to be prepared in advance. When in the main program, switch from regular operation to the flash memory rewrite operation, then execute the flash memory rewrite program you've prepared 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

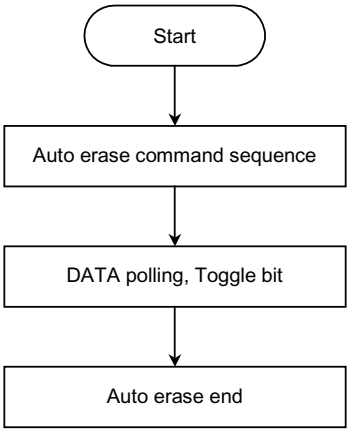
Auto program



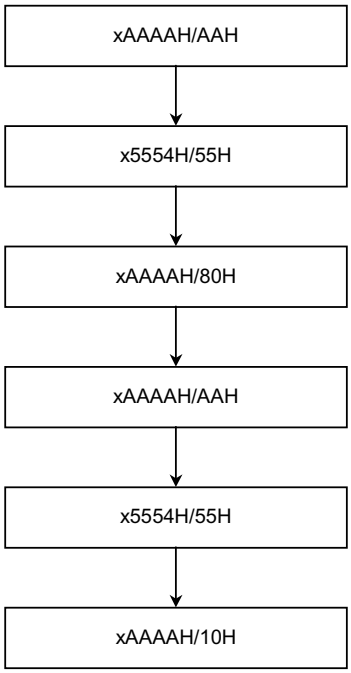
Auto program command sequence (address/command)



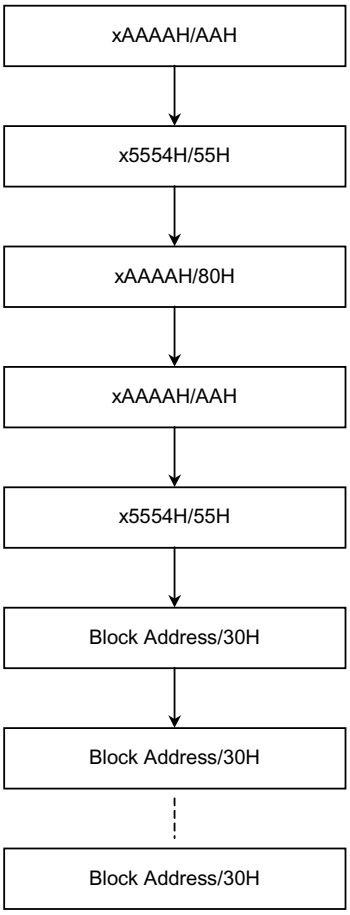
Auto erase



Auto Chip erase command sequence
(Address/Command)

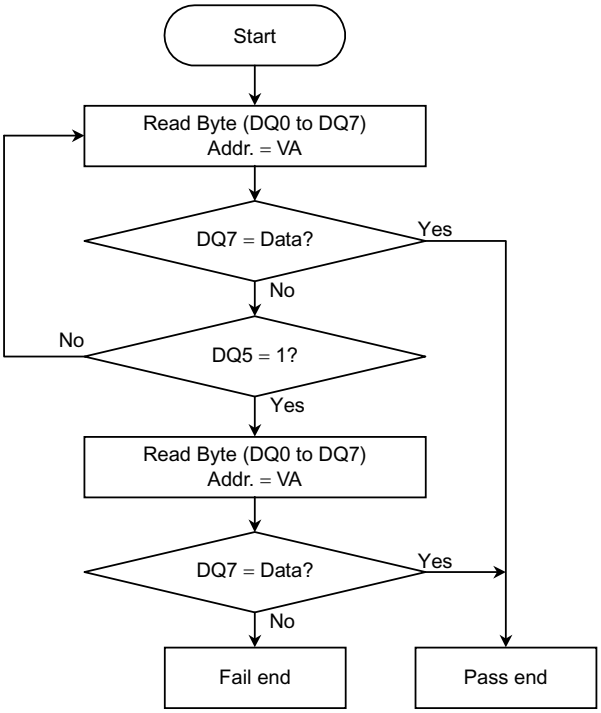


Auto Block/Multi Block erase command sequence
(Address/Command)

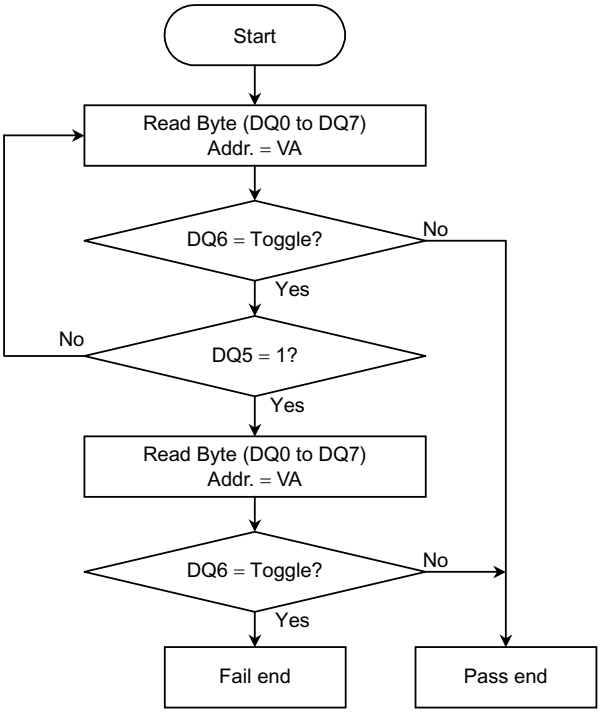


Address input at auto multi block erase (under 50 μs each)

DQ7 DATA polling



DQ6 Toggle bit



VA: Programmed address at auto program.
Flash memory address at auto chip erase.
Selected block address at auto block erase.

3.4 Single Boot Mode

(1) Outline

TMP91FY12A 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 1) 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.

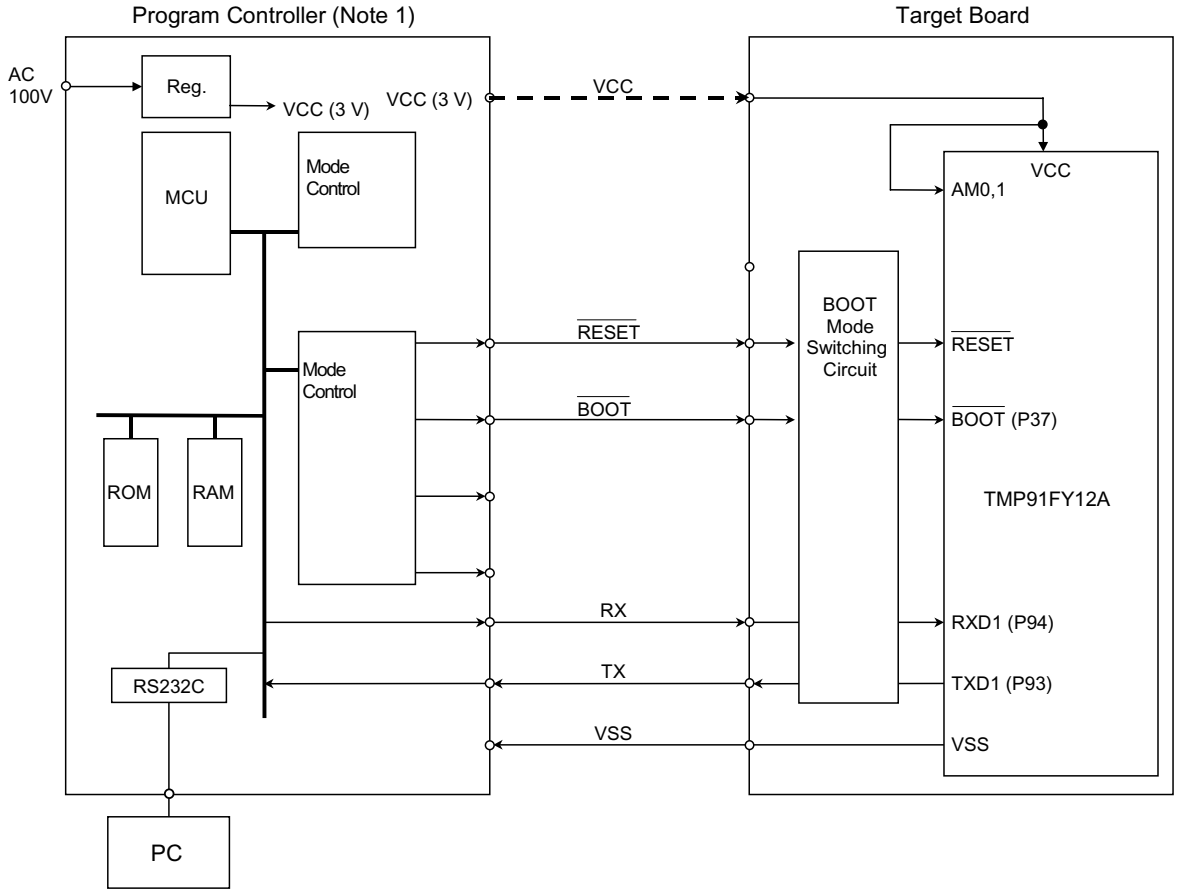


Figure 3.4.1 Example for Connecting Units for On-board Programming

Note: One of the programming controllers supported for the TMP91FY12A is the AF210, AF220, AF110, AF120 (Advanced On-board Flash Microcomputer Programmer) from YDC Co. For details, refer to the manual included with the AF210, AF220, AF110, AF120.

Where to contact: YDC Co.
Instruments Business Division, Instruments Development Center
TEL: 81-42-333-6224

(2) Mode setting

To execute on-board programming, start the TMP91FY12A in Single Boot mode as follows:

AM0	=	H
AM1	=	H
$\overline{\text{BOOT}}$ (P37)	=	L
$\overline{\text{RESET}}$	=	↑

After setting the AM0, AM1, and $\overline{\text{BOOT}}$ pins each to the above conditions, drive the signal input to the $\overline{\text{RESET}}$ pin high. The TMP91FY12A starts up in single-boot mode.

(3) Memory map

Figure 3.4.2 compares memory maps in single-chip and the single-boot modes. When in single boot mode, the internal flash memory is mapped into addresses 1000H through 4FFFFH, as shown here.

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

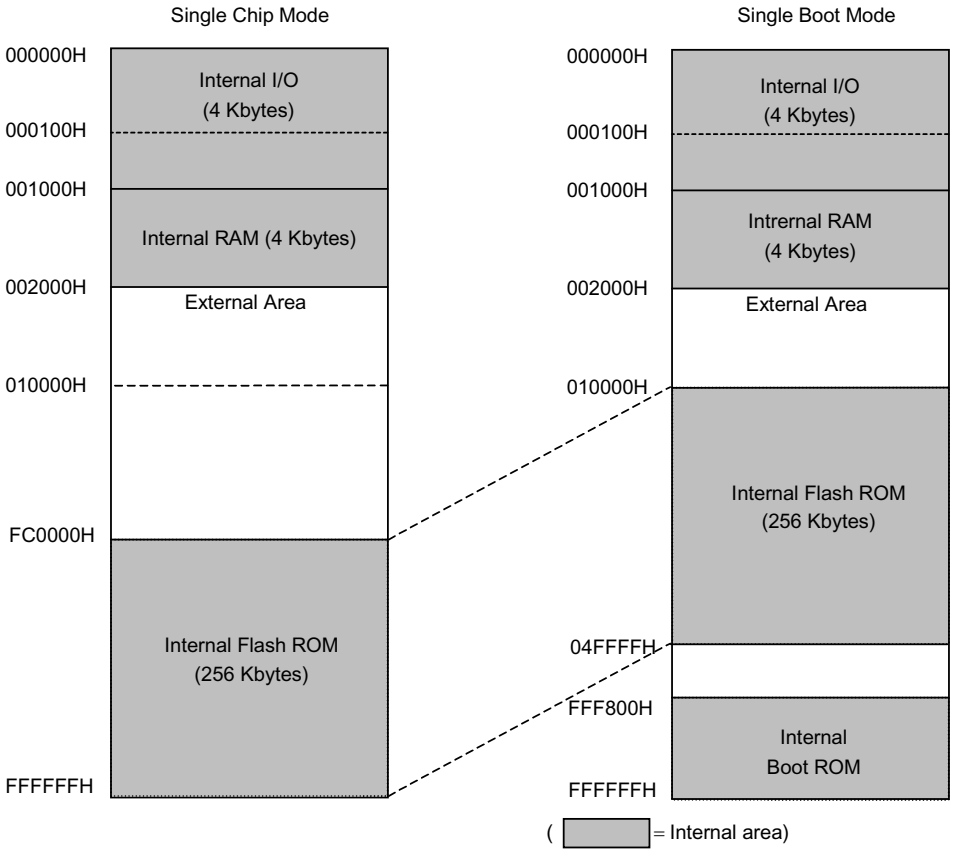


Figure 3.4.2 Comparison of Memory Maps

(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 TMP91FY12A.

Note that although the default baud rate is 9,600 bps, it can be changed to other values as shown in Table 3.4.1.

Communication channel	:	SIO channel 1
Serial transfer mode	:	UART (asynchronous communication) mode, full-duplex communication
Data length	:	8 bits
Parity bit	:	None
Stop bit	:	1 bit
Baud rate (default)	:	9600bps

(5) Data transfer format

Table 3.4.1 through Table 3.4.7 show the 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

Baud rate modification data	04H	05H	06H	07H	0AH	18H	28H
Baud rate (bps)	76800	62500	57600	38400	31250	19200	9600

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

Table 3.4.2 Operating 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: TMP91FY12A

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)	(%)
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 10.20	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	+4.17	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	—	—	—	—	—	—

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 (9600bps) of program controller, oscillator frequency and detecting timing of matching data is under +/−3%.

Table 3.4.4 Boot Program Transfer format (For Flash Memory Rewrite)

	Number of Bytes Transferred	Transfer Data from Controller to TMP91FY12A	Baud rate	Transfer Data from TMP91FY12A to Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) –	9600 bps 9600 bps	– (baud rate auto set) OK: Echoback data (5AH) NG: Nothing Transmitted.
	3rd byte 4th byte	Baud rate modification data (See Table 3.4.1) –	9600 bps 9600 bps	– OK: Echoback data NG: A1H × 3, A2H × 3, A3H × 3 (Note 1)
	5th byte 6th byte	Operation command data (30H) –	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	Expanded Intel Hex format (binary) (Note 2)	Changed new baud rate	–
	n'th – 1 byte	–	Changed new baud rate	OK: SUM (High) NG: Nothing Transmitted. (Note 3)
	n'th byte	–	Changed new baud rate	OK: SUM (Low) NG: Nothing Transmitted. (Note 3)
	n'th + 1 byte	(Wait for next operation command data.)	Changed new baud rate	–

Note 1: xxH × 3 denoes that operation stops after sending 3 bytes of xxH.

Note 2: Refer to the "Notes on Expanded Intel Hex Format (binary)" in the latter page of this manual.

Note 3: Refer to the "Notes on Sum" in the latter page of this manual.

Table 3.4.5 Transfer Format for Boot Program: For RAM Loader

	Number of Bytes Transferred	Transfer Data from Controller to TMP91FY12A	Baud rate	Transfer Data from TMP91FY12A to Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) –	9600bps 9600bps	– (baud rate auto set) OK: Echoback data (5AH) NG: Nothing Transmitted.
	3rd byte 4th byte	Baud rate modification data (See Table 3.4.1) –	9600bps 9600bps	– OK: Echoback data NG: A1H × 3, A2H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (60H) –	Changed new baud rate Changed new baud rate	– OK: Echoback data (30H) NG: A1H × 3, A2H × 3, A3H × 3, 63H × 3
	7th byte 8th byte	Address 23-16 in which to store password count (Note 2) –	Changed new baud rate Changed new baud rate	– OK: Nothing Transmitted. NG: A1H × 3, A2H × 3, A3H × 3 (Note 1)
	9th byte 10th byte	Address 15-08 in which to store password count (Note 2) –	Changed new baud rate Changed new baud rate	– OK: Nothing Transmitted. NG: A1H × 3, A2H × 3, A3H × 3 (Note 1)
	11th byte 12th byte	Address 07-00 in which to store password count (Note 2) –	Changed new baud rate Changed new baud rate	– OK: Nothing Transmitted. NG: A1H × 3, A2H × 3, A3H × 3 (Note 1)
	13th byte 14th byte	Address 23-16 in which to store password comparison –	Changed new baud rate Changed new baud rate	– OK: Nothing Transmitted. NG: A1H × 3, A2H × 3, A3h × 3 (Note 1)
	15th byte 16th byte	Address 15-08 in which to store password comparison (Note 2) –	Changed new baud rate Changed new baud rate	– OK: Nothing Transmitted. NG: A1H × 3, A2H × 3, A3H × 3 (Note 1)
	17th byte 18th byte	Address 07-00 in which to store password comparison (Note 2) –	Changed new baud rate Changed new baud rate	– OK: Nothing Transmitted. NG: A1H × 3, A2H × 3, A3H × 3 (Note 1)
	19th byte : m'th byte	Password string (Note 2)	Changed new baud rate Changed new baud rate	– OK: Nothing Transmitted. NG: A1H × 3, A2H × 3, A3H × 3 (Note 1)
	m'th + 1 byte : n'th – 2 byte	Expanded Intel Hex format (binary) (Note 3)		–
	n'th – 1 byte	–	Changed new baud rate	OK: SUM (High) NG: Nothing Transmitted. (Note 4)
	n'th byte	–	Changed new baud rate	OK: SUM (Low) NG: Nothing Transmitted. (Note 4)
RAM	–	Jump to user program's start address		

Note 1: xxH × 3 denoes that operation stops after sending 3 bytes of xxH.

Note 2: Refer to the "Notes on Password" in the latter page of this manual.

Note 3: Refer to the "Notes on Expanded Intel Hex Format (binary)" in the latter page of this manual.

Note 4: Refer to the "Notes on Sum" in the latter page of this manual.

Table 3.4.6 Transfer Format for Boot Program: For flash memory Sum

	Number of Bytes Transferred	Transfer Data from Controller to TMP91FY12A	Baud rate	Transfer Data from TMP91FY12A to Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) –	9600bps 9600bps	– (baud rate auto set) OK: Echoback data (5AH) NG: Nothing Transmitted.
	3rd byte 4th byte	Baud rate modification data (See Table 3.4.1) –	9600bps 9600bps	– OK: Echoback data NG: A1H × 3, A2H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operating command data (90H) –	Changed new baud rate Changed new baud rate	– OK: Echoback data (90H) NG: A1H × 3, A2H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte	–	Changed new baud rate	OK: SUM (High) (Note 2) NG: –
	8th byte	–	Changed new baud rate	OK: SUM (Low) (Note 2) NG: –
	9th byte	(Wait for next operation command data.)	Changed new baud rate	–

Note 1: xxH × 3 denoes that operation stops after sending 3 bytes of xxH.

Note 2: Refer to the “Notes on SUM.”

(6) Description of boot program operation

When you start the TMP91FY12A in single-boot mode, the boot program starts up. The boot program provides the functions described below.

For details about these functions, refer to A Flash memory rewrite program through C 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.

A. 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 channel's initial baud rate to 9,600 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: 9,600 bps).

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 int 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-2byte 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.

B. 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 Extend 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 of 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.

C. 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 "Note 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.

D. 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	Operating command error occurred.
64H, 64H, 64H	Flash memory erase error occurred.
A1H, A1H, A1H	Framing error in received data occurred. (Note)
A2H, A2H, A2H	Parity error in received data occurred. (Note)
A3H, A3H, A3H	Overrun error in received data occurred. (Note)

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

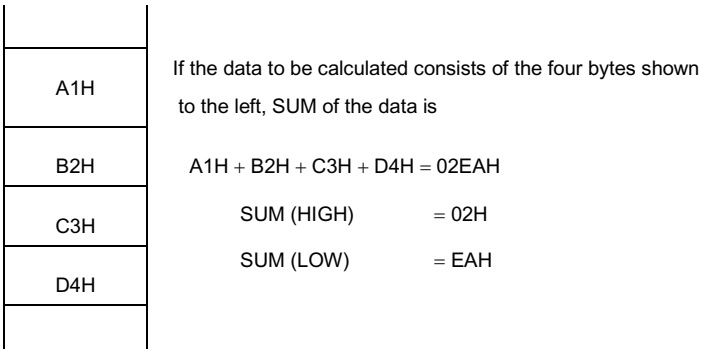
E. Notes on SUM

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:



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

Operating 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	–

F. Notes of 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 TMP91FY12A 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 overwrite command data

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 TMP91FY12A	Extended record	: 02 0000 02 1000 EC zz
Controller to TMP91FY12A	Data record (data length: 08H)	: 08 FFF8 00 xxxxxx CS zz
Controller to TMP91FY12A	Extended record	: 02 0000 02 2000 DC zz
Controller to TMP91FY12A	Data record (data length: 30H)	: 30 0000 00 yyyyyyyy CS zz
Controller to TMP91FY12A	End record	: 00 0000 01 FF ww
TMP91FY12A to Controller	SUM (upper byte) (n'th – 1 byte in Table 3.4.4)	SUM (upper byte)
TMP91FY12A to Controller	SUM (lower byte) (n'th byte in Table 3.4.4)	SUM (lower byte)
Controller to TMP91FY12A	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.

G. 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 cont strage address (PNSA)

The content of the address specified by PNSA is the password cont (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 alldata 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.

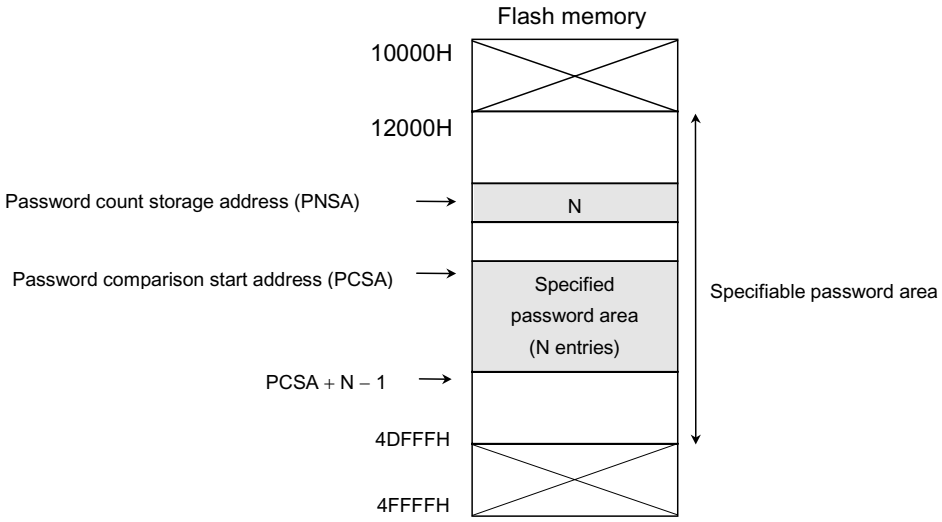
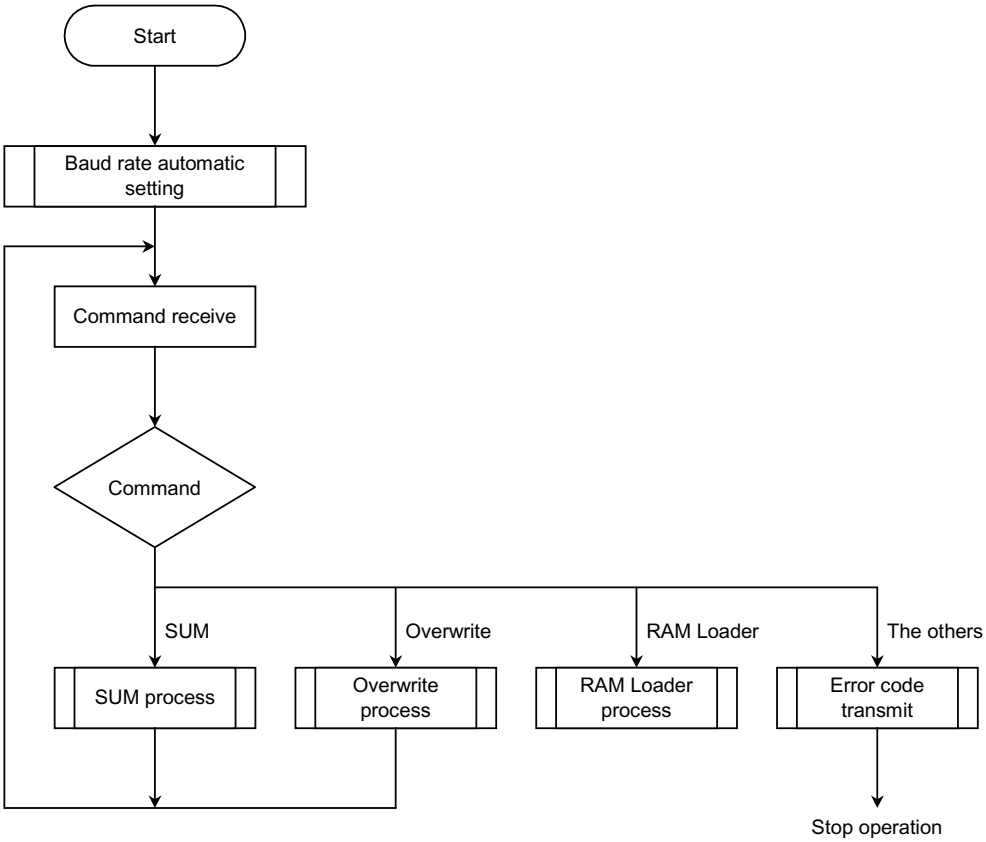
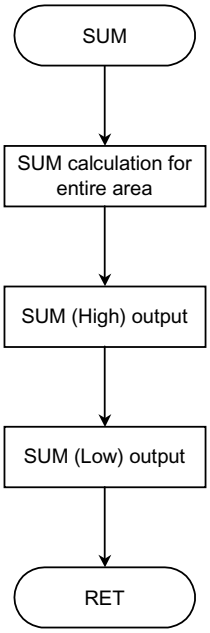


Figure 3.4.3 Conceptual Diagram of Password Area

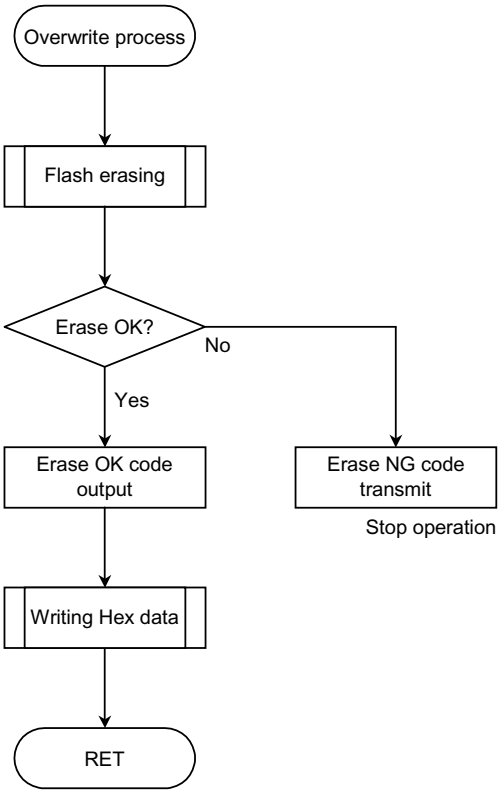
Single Boot General Flow



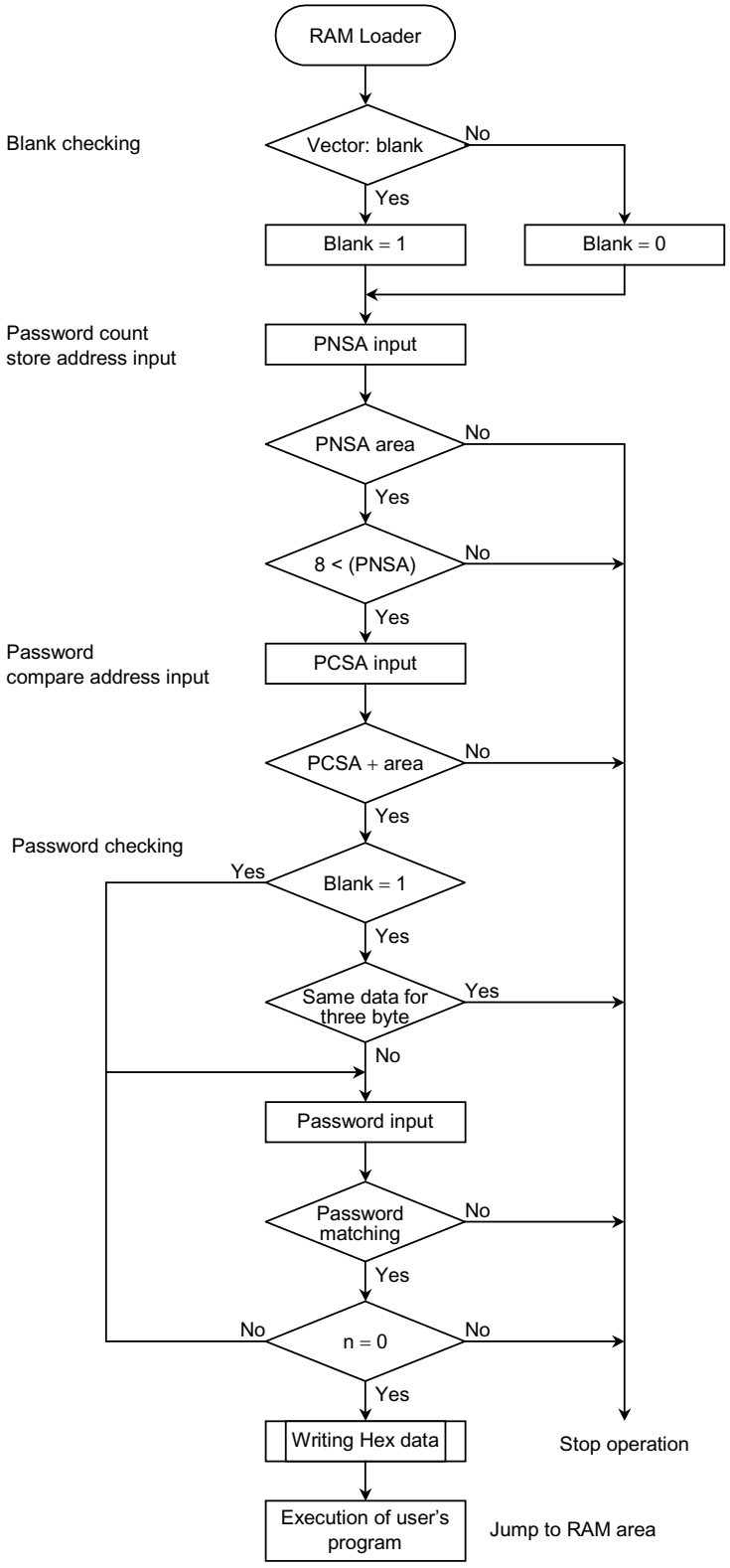
(1) SUM command



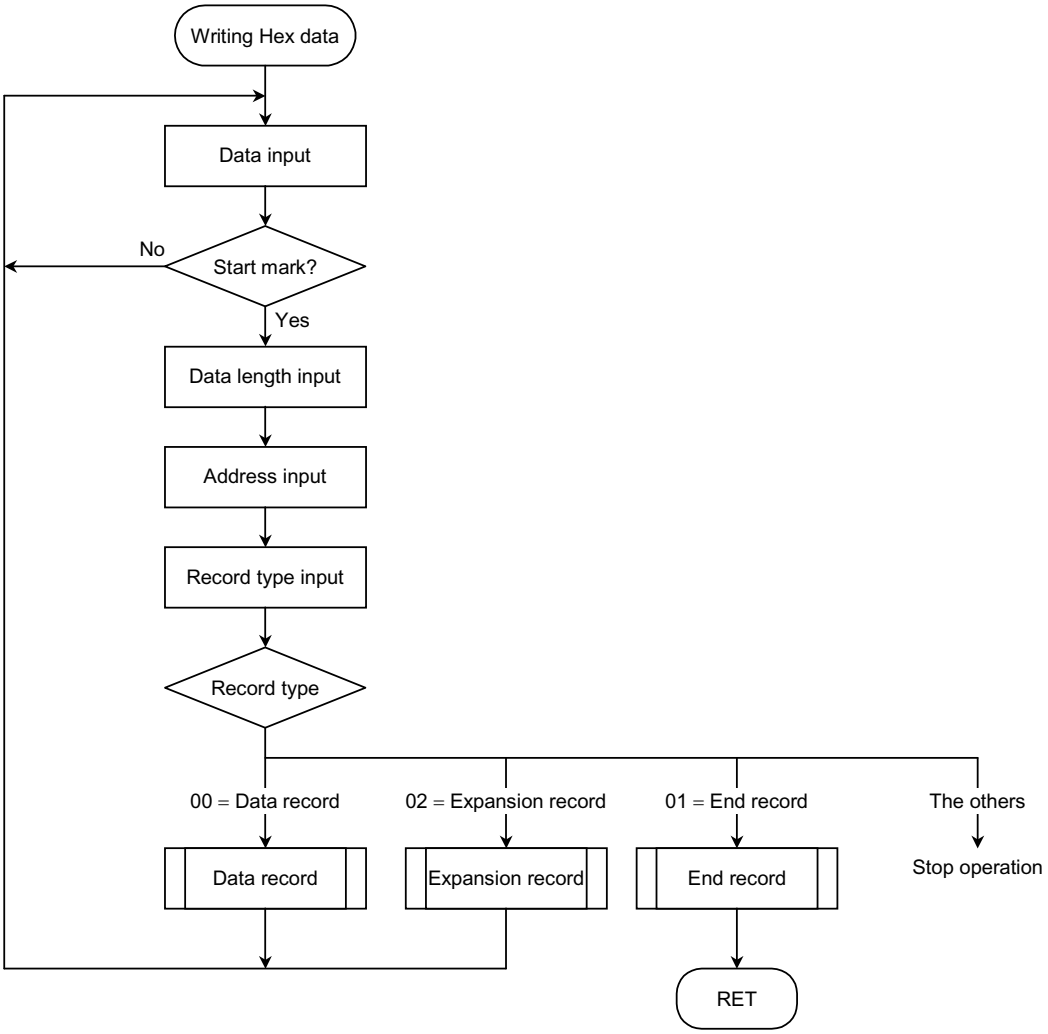
(2) Overwrite command



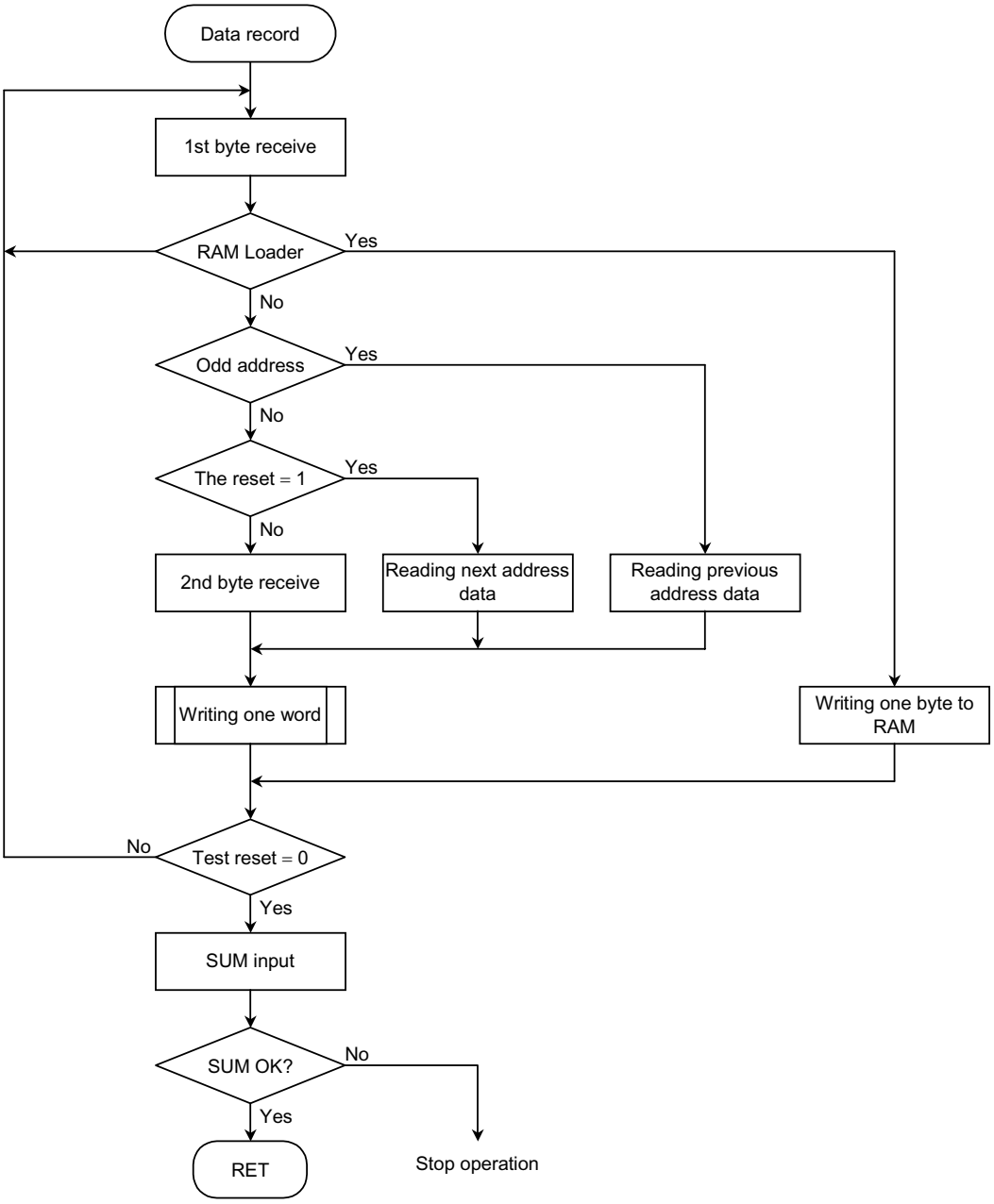
(3) RAM Loader command



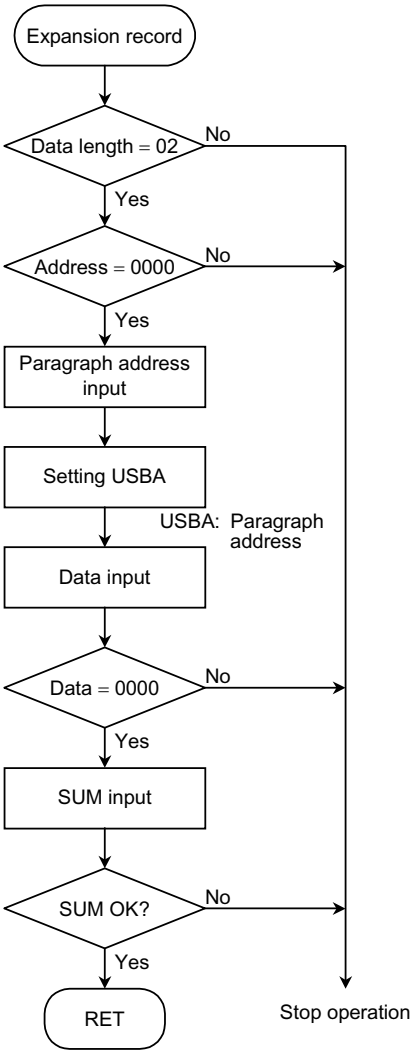
(2) - 1 Writing HEX data



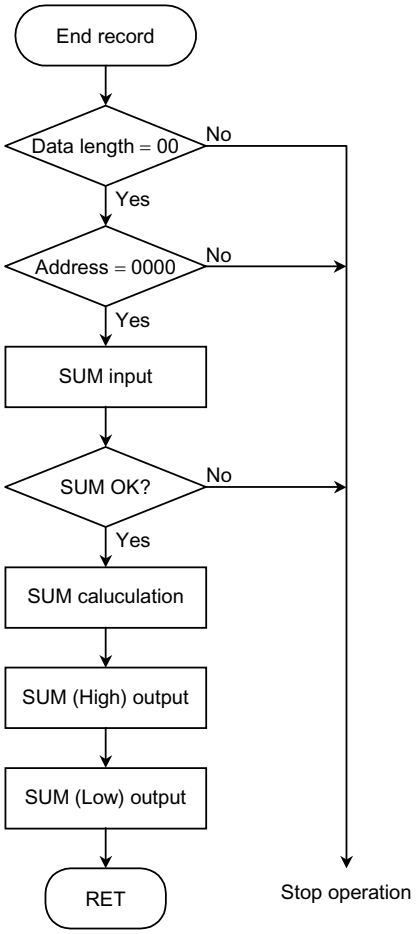
(2) - 1 - 1 Data record



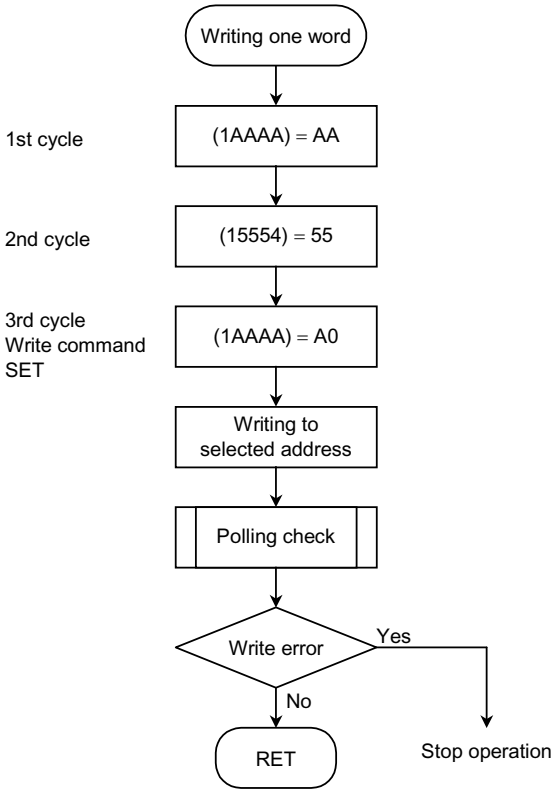
(2) - 1 - 2 Expansion record



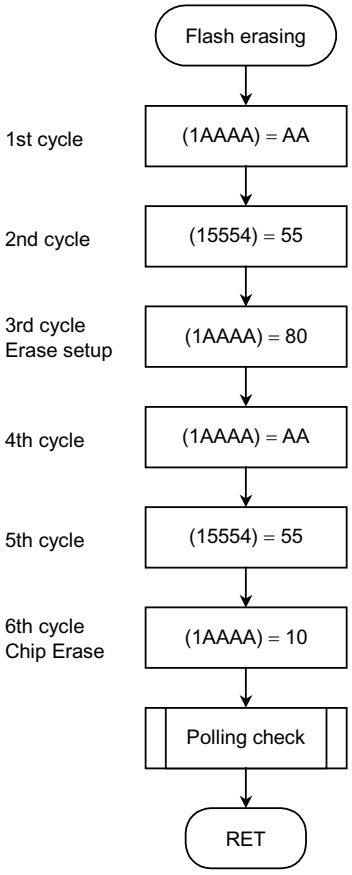
(2) - 1 - 3 End record



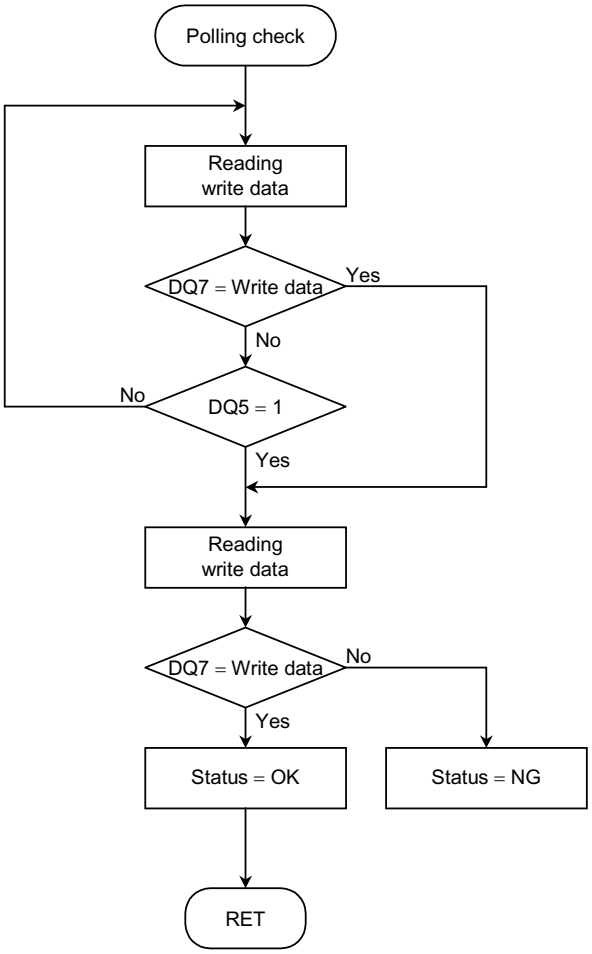
(2) - 1 - 1 - 1 Writing one word



(2) - 2 Erasing Flash memory



(2) - 2 - 1 Data polling



4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power Supply Voltage	Vcc	−0.5 to 4.0	V
Input Voltage	VIN	−0.5 to Vcc + 0.5	V
Output Current	IOL	2	mA
Output Current	IOH	−2	mA
Output Current (total)	ΣIOL	80	mA
Output Current (total)	ΣIOH	−80	mA
Power Dissipation (Ta = 85°C)	PD	600	mW
Soldering Temperature (10 s)	TSOLDER	260	°C
Storage Temperature	TSTG	−65 to 150	°C
Operating Temperature	TOPR	−20 to 70	°C
Number of Times Program Erase	N_{EW}	10000	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 (1/2)

Parameter		Symbol	Condition		Min	Typ. (Note 1)	Max	Unit
Power Supply Voltage (Avcc = DVcc) (Avss = DVss = 0 V)		V_{CC}	fc = 4 to 27 MHz	fs = 30 to 34 kHz	2.7		3.6	V
Input Low Voltage	P00 to P17 (AD0 to 15)	V_{IL}	Vcc = 2.7 to 3.6 V		−0.3		0.6	V
	P20 to PA7 (except P63)	V_{IL1}	Vcc = 2.7 to 3.6 V				0.3 Vcc	
	$\overline{RESET}$, $\overline{NMI}$, P63 (INT0)	V_{IL2}	Vcc = 2.7 to 3.6 V				0.25 Vcc	
	AM0, 1	V_{IL3}	Vcc = 2.7 to 3.6 V				0.3	
	X1	V_{IL4}	Vcc = 2.7 to 3.6 V				0.2 Vcc	
Input High Voltage	P00 to P17 (AD0 to 15)	V_{IH}	Vcc = 2.7 to 3.6 V		2.0		Vcc + 0.3	
	P20 to PA7 (except P63)	V_{IH1}	Vcc = 2.7 to 3.6 V		0.7 Vcc			
	$\overline{RESET}$, $\overline{NMI}$, P63 (INT0)	V_{IH2}	Vcc = 2.7 to 3.6 V		0.75 Vcc			
	AM0, 1	V_{IH3}	Vcc = 2.7 to 3.6 V		Vcc−0.3			
	X1	V_{IH4}	Vcc = 2.7 to 3.6 V		0.8 Vcc			
Output Low Voltage		V_{OL}	IOL = 1.6 mA	Vcc = 2.7 to 3.6 V			0.45	V
Output High Voltage		V_{OH}	IOH = −400 μA	Vcc = 2.7 to 3.6 V	2.4			

Note 1: Typical values are for when Ta = 25°C and Vcc = 3.0 V unless otherwise noted.

4.2 DC Characteristics (2/2)

Parameter	Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
Input Leakage Current)	ILI	$0.0 \leq V_{IN} \leq Vcc$		0.02	±5	μA
Output Leakage Current	ILO	$0.2 \leq V_{IN} \leq Vcc - 0.2$		0.05	±10	
Power Down Voltage (at STOP, RAM back-up)	VSTOP	V IL2 = 0.2 Vcc, V IH2 = 0.8 Vcc	2.7		3.6	V
$\overline{RESET}$ Pull-up Resistor	RRST	Vcc = 3 V ± 10%	100		400	KΩ
Pin Capacitance	CIO	fc = 1 MHz			10	PF
Schmitt Width $\overline{RESET}$, $\overline{NMI}$, INT0	VTH	Vcc = 2.7 to 3.6 V	0.4	1.0		V
Programmable Pull-up Resistor	RKH	Vcc = 3 V ± 10%	100		400	KΩ
Normal (Note 2)	Icc	Vcc = 3 V ± 10% fc = 27 MHz		30.0	45.0	mA
Idle2				4.5	7.0	
Idle1				2.0	4.0	
Slow (Note 2)		Vcc = 3 V ± 10% fs = 32.768 kHz		30.0	40	μA
Idle2				9.0	25	
Idle1				6.0	15	
Stop		Vcc = 3 V ± 10%		1.0	15	μA

Note 1: Typical values are for when Ta = 25°C and Vcc = 3.0 V unless otherwise noted.

Note 2: Icc measurement conditions (Normal, Slow):
All functions are operating; output pins are open and input pins are fixed.

4.3 AC Characteristics

(1) Vcc = 3.0 V ± 10%

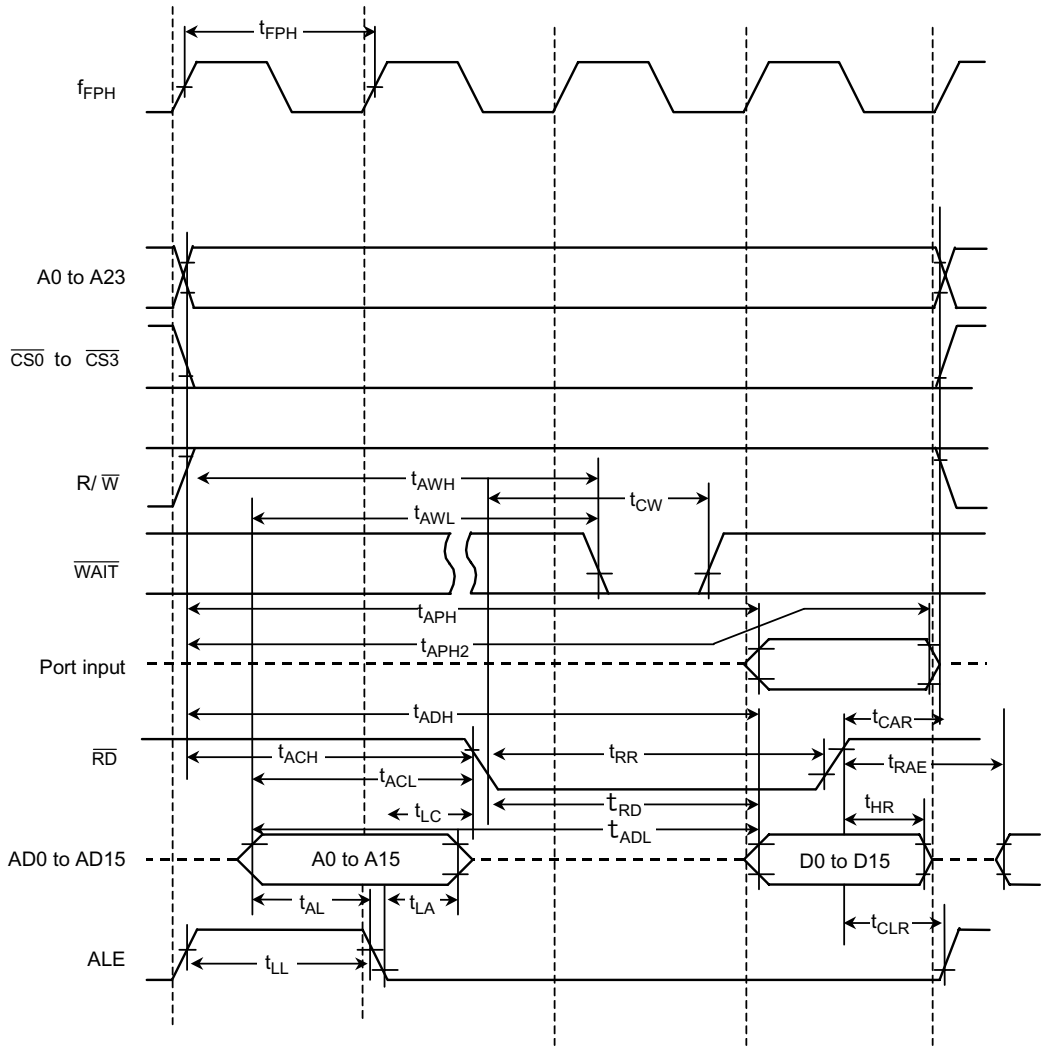
No.	Parameter	Symbol	Variable		f_{FPH} = 27 MHz		Unit
			Min	Max	Min	Max	
1	f_{FPH} Period (= x)	t_{FPH}	37.0	31250	37.0		ns
2	A0 to A15 Vaild → ALE Fall	t_{AL}	0.5x – 6		12		ns
3	ALE Fall → A0 to A15 Hold	t_{LA}	0.5x – 16		2		ns
4	ALE High Width	t_{LL}	x – 20		17		ns
5	ALE Fall → $\overline{RD}$ / $\overline{WR}$ Fall	t_{LC}	0.5x – 14		4		ns
6	$\overline{RD}$ Rise → ALE Rise	t_{CLR}	0.5x – 10		8		ns
7	$\overline{WR}$ Rise → ALE Rise	t_{CLW}	x – 10		27		ns
8	A0 to A15 Valid → $\overline{RD}$ / $\overline{WR}$ Fall	t_{ACL}	x – 23		14		ns
9	A0 to A23 Valid → $\overline{RD}$ / $\overline{WR}$ Fall	t_{ACH}	1.5x – 26		29		ns
10	$\overline{RD}$ Rise → A0 to A23 Hold	t_{CAR}	0.5x – 13		5		ns
11	$\overline{WR}$ Rise → A0 to A23 Hold	t_{CAW}	x – 13		24		ns
12	A0 to A15 Valid → D0 to D15 Input	t_{ADL}		3.0x – 38		73	ns
13	A0 to A23 Valid → D0 to D15 Input	t_{ADH}		3.5x – 41		88	ns
14	$\overline{RD}$ Fall → D0 to D15 Input	t_{RD}		2.0x – 30		44	ns
15	$\overline{RD}$ Low Width	t_{RR}	2.0x – 15		59		ns
16	$\overline{RD}$ Rise → D0 to A15 Hold	t_{HR}	0		0		ns
17	$\overline{RD}$ Rise → A0 to A15 Output	t_{RAE}	x – 15		22		ns
18	$\overline{WR}$ Low Width	t_{WW}	1.5x – 15		40		ns
19	D0 to D15 Valid → $\overline{WR}$ Rise	t_{DW}	1.5x – 35		20		ns
20	$\overline{WR}$ Rise → D0 to D15 Hold	t_{WD}	x – 25		12		ns
21	A0 to A23 Valid → $\overline{WAIT}$ Input (1 WAIT +n Mode)	t_{AWH}		3.5x – 60		69	ns
22	A0 to A15 Valid → $\overline{WAIT}$ Input (1 WAIT +n Mode)	t_{AWL}		3.0x – 50		61	ns
23	$\overline{RD}$ / $\overline{WR}$ Fall → $\overline{WAIT}$ Hold (1 WAIT +n Mode)	t_{CW}	2.0x + 0		74		ns
24	A0 to A23 Valid → Port Input	t_{APH}		3.5x – 89		40	ns
25	A0 to A23 Valid → Port Hold	t_{APH2}	3.5x		129		ns
26	A0 to A23 Valid → Port Valid	t_{AP}		3.5x + 80		209	ns

AC Measuring Conditions

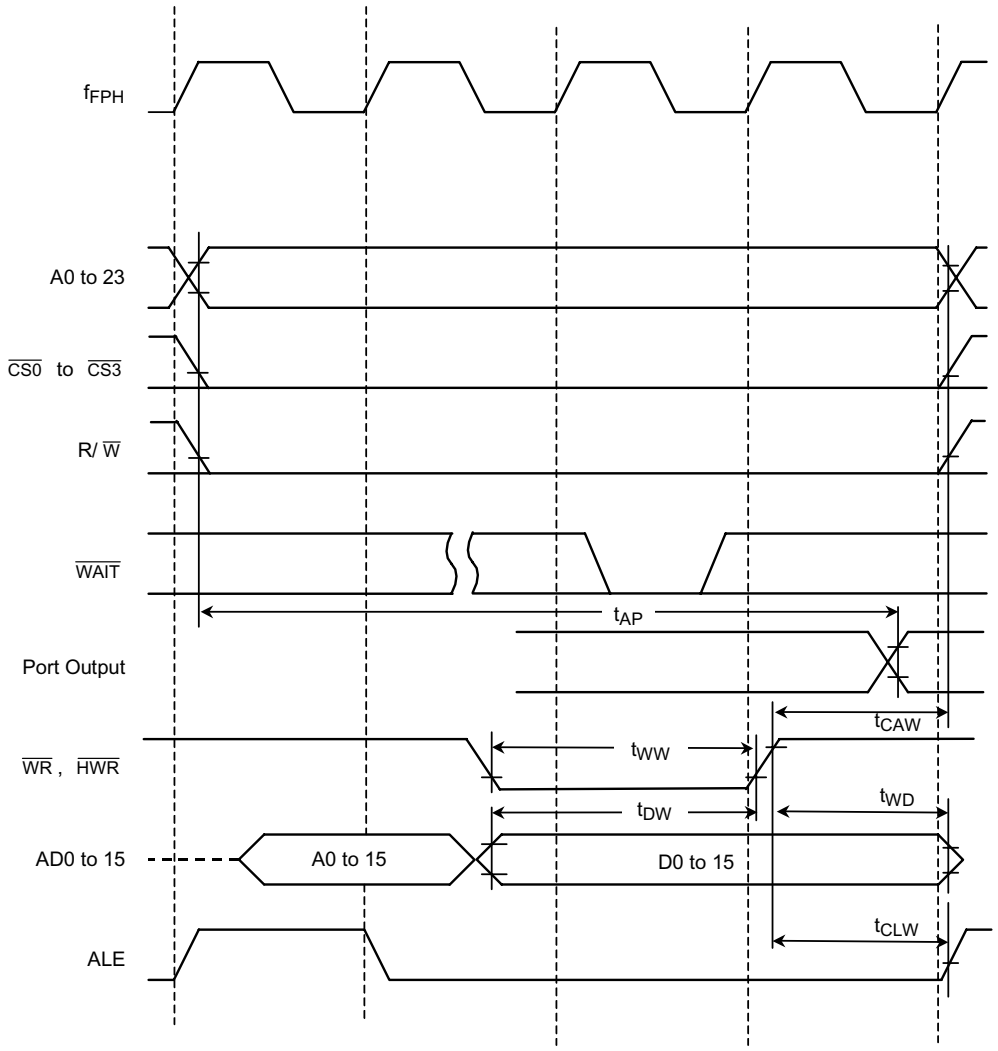
- Output Level: High = 0.7 × Vcc, Low = 0.3 × Vcc, CL = 50 pF
- Input Level: High = 0.9 × Vcc, Low = 0.1 × Vcc

Note: x used in an expression shows a frequency for the clock f_{FPH} selected by SYSCR1<SYSCK>.
The value of x changes according to whether a clock gear or a low-speed oscillator is selected.
An example value is calculated for fc, with gear = 1/fc (SYSCR1<SYSCK, GEAR2 to 0> = 0000).

(2) Raed Cycle


tFPH
fFPH
A0 to A23
CS0 to CS3
R/W
tAWH
tCW
tAWL
WAIT
tAPH
tAPH2
Port input
tADH
tCAR
RD
tACH
tRR
tRAE
tACL
tLC
tRD
tHR
tADL
AD0 to AD15
A0 to A15
D0 to D15
tAL
tLA
tCLR
ALE
tLL

(3) Write Cycle



4.4 AD Conversion Characteristics

AVcc = Vcc, AVss = Vss

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog Reference Voltage (+)	VREFH	V_{CC} = 3 V ± 10%	V_{CC} − 0.2 V	V_{CC}	V_{CC}	V
Analog Reference Voltage (−)	VREFL	V_{CC} = 3 V ± 10%	V_{SS}	V_{SS}	V_{SS} + 0.2 V	
Analog Input Voltage Range	VAIN		V_{REFL}		V_{REFH}	
Analog Current for Analog Reference Voltage <VREFON> = 1	IREF (VREFL = 0 V)	V_{CC} = 3 V ± 10%		0.94	1.20	mA
<VREFON> = 0		V_{CC} = 3 V ± 10%		0.02	5.0	μA
Error (not including quantizing errors)	−	V_{CC} = 3 V ± 10%		±1.0	± 4.0	LSB

Note 1: 1 LSB = (VREFH − VREFL)/1024 [V]

Note 2: The operation above is guaranteed for $f_{FPH} \geq 4$ MHz.

Note 3: The value for I_{CC} includes the current which flows through the AVCC pin.

4.5 Serial Channel Timing (I/O Internal Mode)

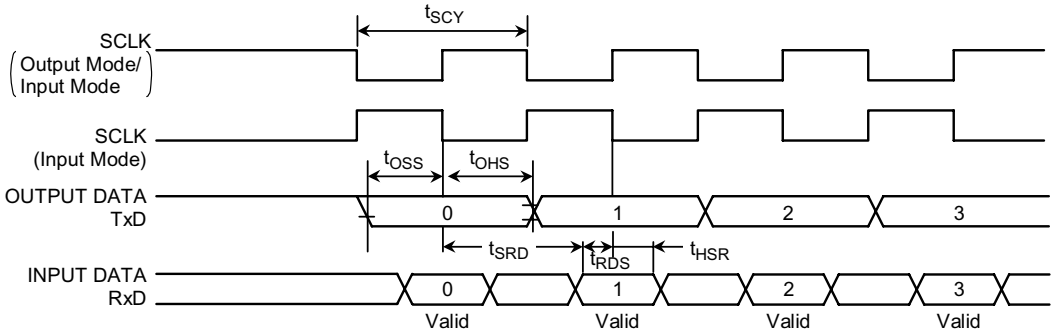
(1) SCLK Input Mode

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Period	t_{SCY}	16X		1.6		0.59		μs
Output Data → SCLK Rising /Falling Edge (Note)	t_{OSS}	$t_{SCY}/2 - 4X\text{-}110$		290		38		ns
SCLK Rising/Falling Edge (Note) → Output Data Hold	t_{OHS}	$t_{SCY}/2 + 2x + 0$		1000		370		ns
SCLK Rising/Falling Edge (Note) → Input Data Hold	t_{HSR}	$3x + 10$		310		121		ns
SCLK Rising/Falling Edge (Note) → Valid Data Input	t_{SRD}		$t_{SCY} - 0$		1600		592	ns
Valid Data InputSCLK →Rising/Falling Edge (Note)	t_{RDS}	0		0		0		ns

Note: SCLK Rinsing/Falling Edge: The rising edge is used in SCLK Rising Mode.
The falling edge is used in SCLK Falling Mode.

(2) SCLK Output Mode

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Period (programable)	t_{SCY}	16X	8192X	1.6	819	0.59	303	μs
Output Data →SCLK Rising Edge	t_{OSS}	$t_{SCY}/2 - 40$		760		256		ns
SCLK Rising Edge → Output Data Hold	t_{OHS}	$t_{SCY}/2 - 40$		760		256		ns
SCLK Rising Edge → Input Data Hold	t_{HSR}	0		0		0		ns
SCLK Rising Edge → Valid Data Input	t_{SRD}		$t_{SCY} - 1x - 180$		1320		375	ns
Valid Data Input → SCLK Rising Edge	t_{RDS}	$1x + 180$		280		217		ns



4.6 Event Counter (TA0IN, TA4IN, TB0IN0, TB0IN1, TB1IN0, TB1IN1)

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock Perild	t_{VCK}	8X + 100		900		396		ns
Clock Low Level Width	t_{VCKL}	4X + 40		440		188		ns
Clock High Level Width	t_{VCKH}	4X + 40		440		188		ns

4.7 Interrupt and Capture

(1) $\overline{NMI}$, INT0 to INT4 Interrupts

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 to INT4 Low level width	t_{INTAL}	4X + 40		440		188		ns
$\overline{NMI}$, INT0 to INT4 High level width	t_{INTAH}	4X + 40		440		188		ns

(2) INT5 to INT8 Interrupts, Capture

The INT5 to INT8 input width depends on the system clock and prescaler clock settings.

System Clock Selected <SYSCK>	Prescaler Clock Selected <PRCK1, PRCK0>	t_{INTBL} (INT5 to INT8 Low level Width)		t_{INTBH} (INT5 to INT8 High Level Width)		Unit
		Variable	f_{FPH} = 27 MHz	Variable	f_{FPH} = 27 MHz	
		Min	Min	Min	Min	
0 (fc)	00 (f_{FPH})	8X + 100	396	8X + 100	396	ns
	10 (fc/16)	128Xc + 0.1	4.8	128Xc + 0.1	4.8	μs
1 (fs)	00 (f_{FPH})	8X + 0.1	244.3	8X + 0.1	244.3	

Note: Xc = Period of Clock fc

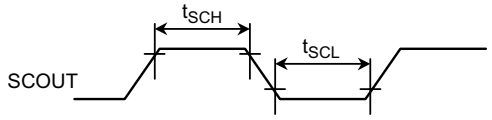
4.8 SCOUT Pin AC Characteristics

Parameter	Symbol	Variable		10 MHz		27 MHz		Condition	Unit
		Min	Max	Min	Max	Min	Max		
Low level Width	t_{SCH}	0.5T – 13		37		5		Vcc = 2.7 to 3.6 V	ns
High level Width	t_{SCL}	0.5T – 13		37		5		Vcc = 2.7 to 3.6 V	ns

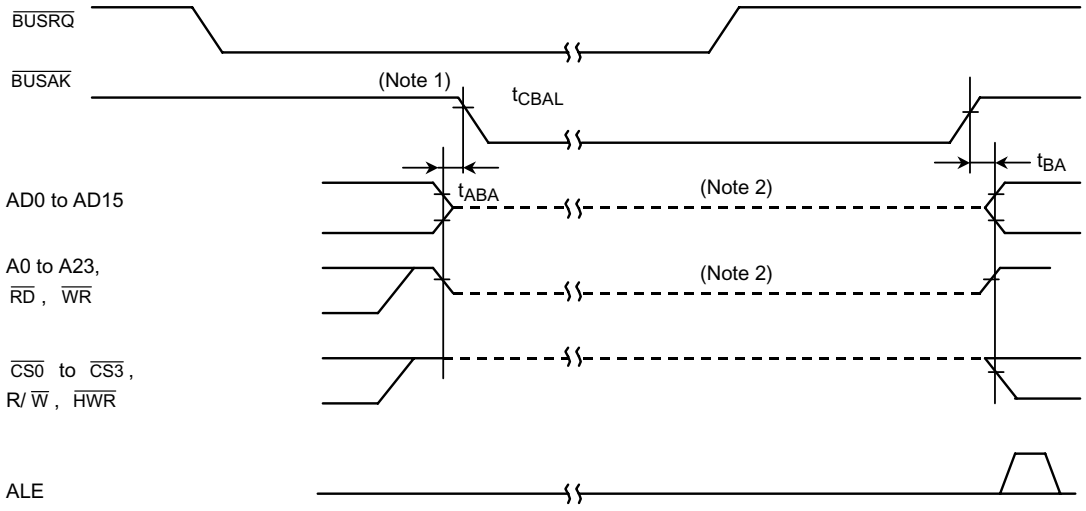
Note: T = Period of SCOUT

Measrement Condition

- Output Level: High 0.7 Vcc/Low 0.3 Vcc, CL = 10 pF



4.9 Bus Request/Bus Acknowledge



Paramter	Symbol	Variable		f_{FPH} = 10 MHz		f_{FPH} = 27 MHz		Condition	Unit
		Min	Max	Min	Max	Min	Max		
Output Buffer Off to $\overline{BUSAK}$ Low	t_{ABA}	0	80	0	80	0	80	Vcc = 2.7 to 3.6 V	ns
$\overline{BUSAK}$ High to Output Buffer On	t_{BAA}	0	80	0	80	0	80	Vcc = 2.7 to 3.6 V	ns

Note 1: Even if the $\overline{BUSRQ}$ Signal goes Low, the bus will not be released while the $\overline{WAIT}$ signal is Low. The bus will only be released when $\overline{BUSRQ}$ goes Low while $\overline{WAIT}$ is High.

Note 2: This line shows only that the output buffer is in the Off state.
It does not indicate that the signal level is fixed.
Just after the bus is released, the signal level set before the bus was released is maintained dynamically by the external capacitance. Therefore, to fix the signal level using an external resister during bus release, careful design is necessary, since fixing of the level is delayed.
The internal programmable pull-up/pull-down resistor is switched between the Active and Non-Active states by the internal signal.

4.10 Recommended Oscillation Circuit

The TMP91FY12AF has been evaluated by the following resonator manufacturer. The evaluation results are shown below for your information.

Note: The load capacitance of the oscillation terminal is the sum of the load capacitances of C1 and C2 to be connected and the stray capacitance on the board. Even if the ratings of C1 and C2 are used, the load capacitance varies with each board and the oscillator may malfunction. Therefore, when designing a board, make the pattern around the oscillation circuit shortest. It is recommended that final evaluation of the resonator be performed on the board.

(1) Examples of resonator connection

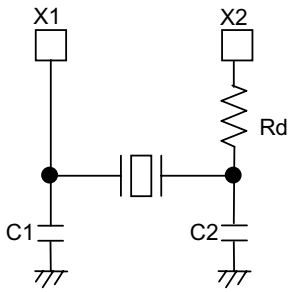


Figure 4.10.1 High-frequency Oscillator Connection

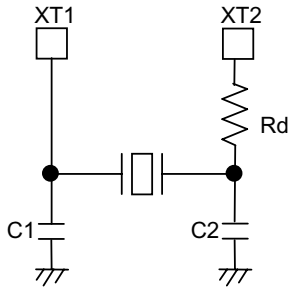


Figure 4.10.2 Low-frequency Oscillator Connection

(2) Recommended ceramic resonators for the TMP91FY12AF: Murata Manufacturing Co., Ltd.

Ta = –40 to 85°C

Item	Oscillation frequency [MHz]	Recommended resonator	Recommended rating			VCC [V]	Remarks
			C1 [pF]	C2 [pF]	Rd [kΩ]		
High-frequency oscillator	4.0	CSTS0400MG06	(47)	(47)	0	2.7 to 3.3	—
	6.75	CSTS0675MG06	(47)	(47)			
	12.5	CSA12.5MTZ	30	30			
		CST12.5MTW	(30)	(30)			
	20.0	CSA20.00MXZ040	7	7			
	27.0	CSA27.00MXZ040	5	5			
		CST27.00MXW040	(5)	(5)			

- The values enclosed in brackets in the C1 and C2 columns apply to the condenser built-in type.
- The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL;
  http://www.murata.co.jp/search/index.html