CMOS 32-bit Micro-controller

TMP94FD53F

Preliminary

1. Outline and Device Characteristics

TMP94FD53 is high-speed advanced 32-bit micro-controller developed for controlling equipment which processes mass data.

TMP94FD53 is a micro-controller which has a high-performance CPU (900/H2 CPU) and various built-in I/Os. TMP94FD53 is housed in a 100-pin mini flat package.

Device characteristics are as follows:

(1) CPU : 32-bit CPU(900/H2 CPU)
- Compatible with TLCS-900,900/L,900/L1,900/H's instruction code
- 16Mbytes of linear address space
- General-purpose register and register banks
- Micro DMA : 8channels (250ns / 4bytes at fc = 20MHz, best case)

(2) Minimum instruction execution time : 50ns(at 20MHz)
- Internal data bus : 32-bit

(3) Internal memory
- Internal RAM : 16K-byte
- Internal ROM : 512K-byte Flash E2PROM
  3K-byte Mask ROM (for Flash boot mode)

(4) External memory expansion
- 16M-byte linear address space (memory mapped I/O)
- External data bus : 8bit(for external I/O expansion)
- * Can't use upper address bus when built-in I/Os are selected

(5) Memory controller
- Chip select output : 1 channel

(6) 8-bit timer : 8 channels
- 8-bit interval timer mode (8 channels)
- 16-bit interval timer mode (4 channels)
- 8-bit programmable pulse generation (PPG) output mode (4 channels)
- 8-bit pulse width modulation (PWM) output mode (4 channels)

(7) 16-bit timer : 2 channels
- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) output mode
- Frequency measurement mode
- Pulse width measurement mode
- Time differential measurement mode

(8) Serial interface : 2 channels
- I/O interface mode
- Universal asynchronous receiver transmitter (UART) mode

(9) Serial expansion interface : 2 channels
- Baud rate 8/4/2/0.5Mbps at fc=20MHz.

(10) Serial bus interface : 2 channels
- Clocked-synchronous 8-bit serial interface mode
- I^2C bus mode

(11) Can controller : 1channel
- Supports CAN version 2.0B.
- 16 mailboxes

(12) 10-bit A/D converter : 12 channels
- A/D conversion time 8μsec @fc=20MHz.
- Total tolerance ±3LSB (excluding quantization error)
- Scan mode for all 12channels

(13) Watch dog timer

(14) Interrupt controller
- 35 internal interrupts
- 9 external interrupts

(15) I/O Port : 70pins

(16) Power supply voltage
- VCC5 = 5V±10% (4.5V to 5.5V)
- VCC3 = 3.3V±0.3V (3V to 3.6V)

(17) Operating temperature : -40℃ to 85℃

(18) Package : P-LQFP100-1414-0.50C

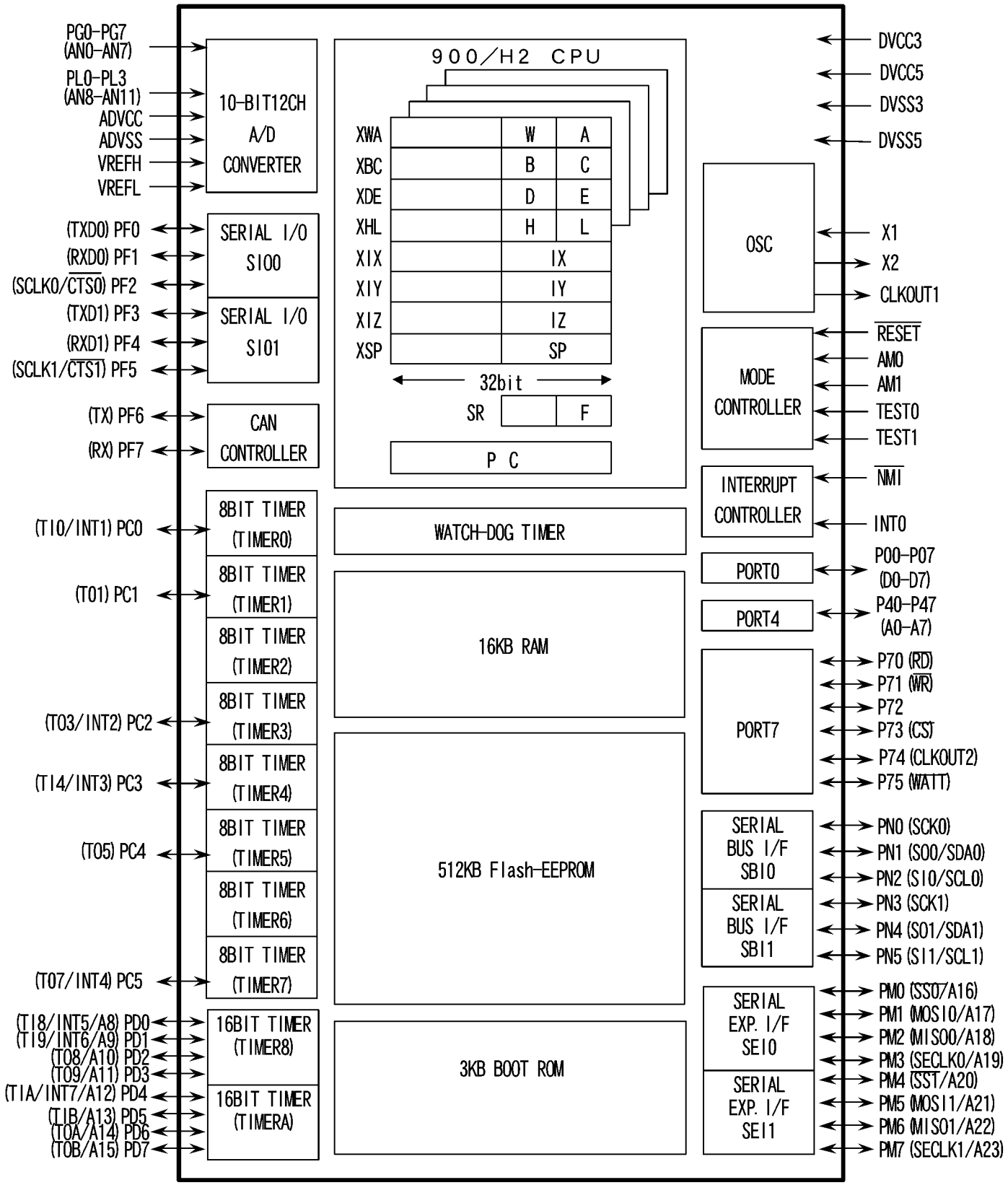


Figure 1 TMP94FD53 block diagram

2. Pin Assignment and Functions

2.1 Pin Assignment

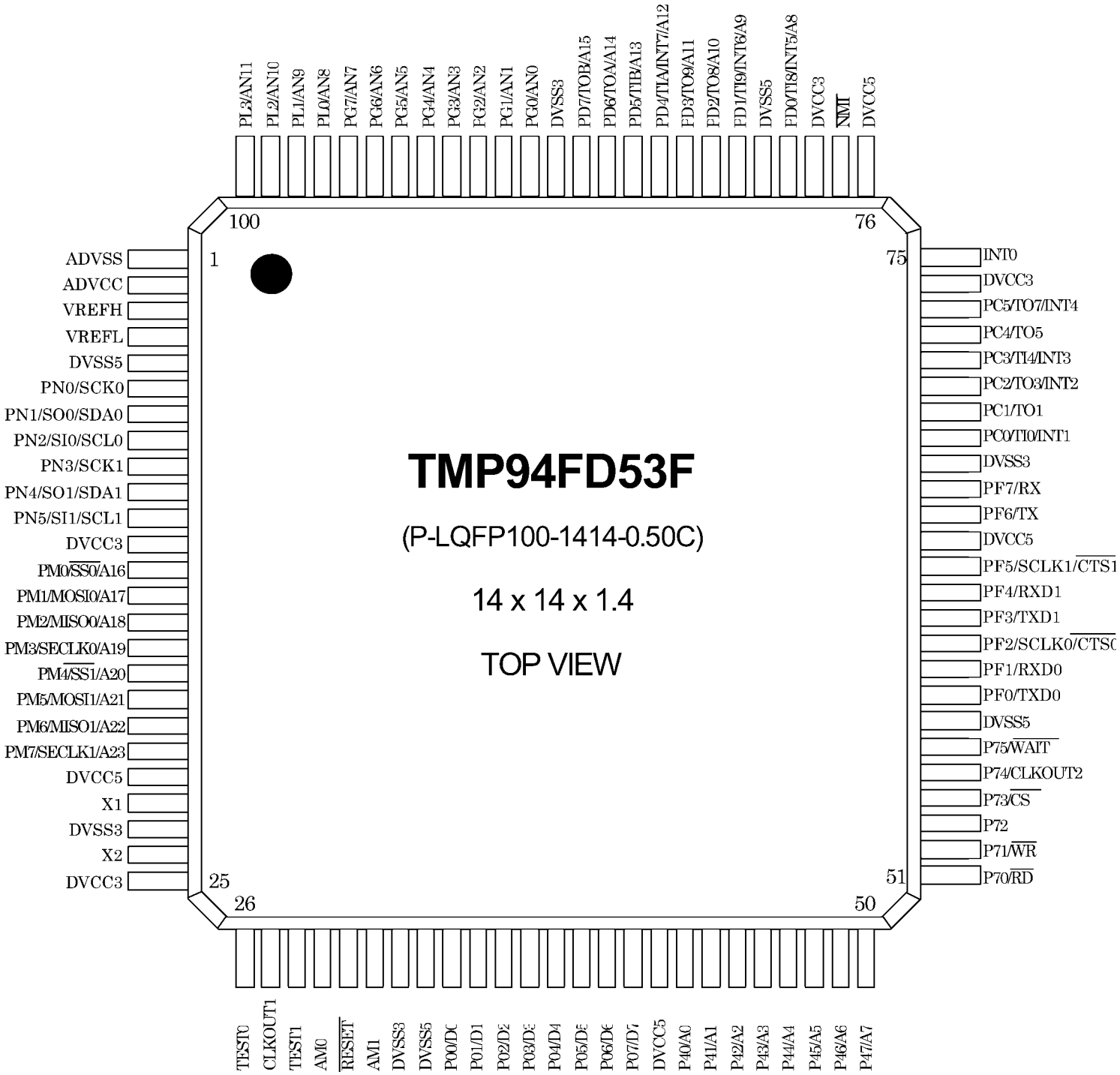


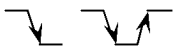
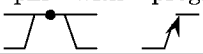
Figure 2.1 Pin Assignment

2.2 Pin names and functions

The following table shows the names and functions of the input/output pins.

Pin Name	Number of pins	In/Out	Function
P00..P07 D0..D7	8 (CMOS) (TTL)	in/out in/out	Port 0: I/O port. Input or output specifiable in units of bits. Data: Data bus 0 to 7.
P40..P47 A0..A7	8	in/out out	Port4: I/O port. Input or output specifiable in units of bits. Address: Address bus 0 to 7.
P70 /RD	1	in/out out	Port70: I/O port. Read: Outputs strobe signal to read external memory.
P71 /WR	1	in/out out	Port 71: I/O port. Write: Output strobe signal to write data on pins.
P72	1	in/out	Port 72: I/O port.
P73 /CS	1	in/out out	Port 73: I/O port. Chip select: Outputs "low" if address is within specified address area.
P74 CLKOUT2	1	in/out out	Port 74: I/O port. Clock output 2: CLKOUT2 output 4 MHz clock at fc = 20 MHz.
P75 /WAIT	1	in/out in	Port 75: I/O port. Wait: Signal used to request CPU bus wait.
PC0 TI0 INT1	1	in/out in in	Port C0: I/O port. Timer input 0: Input pin for timer 0. Interrupt request pin 1: Rising-edge interrupt request pin.
PC1 TO1	1	in/out out	Port C1: I/O port. Timer output 1: Output pin for timer 1.
PC2 TO3 INT2	1	in/out out in	Port C2: I/O port. Timer output 3: Output pin for timer 3. Interrupt request pin 2: Rising-edge interrupt request pin.
PC3 TI4 INT3	1	in/out in in	Port C3: I/O port. Timer input 4: Input pin for timer 4. Interrupt request pin 3: Rising-edge interrupt request pin.
PC4 TO5	1	in/out out	Port C4: I/O port. Timer output 5: Output pin for timer 5.
PC5 TO7 INT4	1	in/out out in	Port C5: I/O port. Timer output 7: Output pin for timer 7. Interrupt request pin 4: Rising-edge interrupt request pin.
PD0 TI8 INT5 A8	1	in/out in in out	Port D0: I/O port. Timer input 8: Input pin for timer 8. Interrupt request pin 5: Interrupt request pin with programmable rising/falling edge. Address: Address bus 8.
PD1 TI9 INT6 A9	1	in/out in in out	Port D1: I/O port. Timer input 9: Input pin for timer 9. Interrupt request pin 6: interrupt request pin with programmable rising/falling edge. Address: Address bus 9.

Pin Name	Number of pins	In/Out	Function
PD2 TO8 A10	1	in/out out out	Port D2: I/O port. Timer output 8: Output pin for timer 8 Address: Address bus 10.
PD3 TO9 A11	1	in/out out out	Port D3: I/O port. Timer output 9: Output pin for timer 9 Address: Address bus 11.
PD4 TIA INT7 A12	1	in/out in in out	Port D4: I/O port. Timer input A: Input pin for timer A Interrupt request pin 7: Interrupt request pin with programmable rising/falling edge. Address: Address bus 12.
PD5 TIB A13	1	in/out in out	Port D5: I/O port. Timer input B: Input pin for timer B. Address: Address bus 13.
PD6 TOA A14	1	in/out out out	Port D6: I/O port. Timer output A: Output pin for timer A. Address: Address bus 14.
PD7 TOB A15	1	in/out out out	Port D7: I/O port. Timer output B: Output pin for timer B. Address: Address bus 15.
PF0 TXD0	1	in/out out	Port F0: I/O port. Serial transmission data 0.
PF1 RXD0	1	in/out in	Port F1: I/O port. Serial receive data 0.
PF2 SCLK0 /CTS0	1	in/out in/out in	Port F2: I/O port. Serial clock input/output 0. Serial data ready to send 0. (Clear-to-send)
PF3 TXD1	1	in/out out	Port F3: I/O port. Serial transmission data 1.
PF4 RXD1	1	in/out in	Port F4: I/O port. Serial receive data 1.
PF5 SCLK1 /CTS1	1	in/out in/out in	Port F5: I/O port. Serial clock input/output 1. Serial data ready to send 0. (Clear-to-send)
PF6 TX	1	in/out out	Port F6: I/O port. CAN transmission data.
PF7 RX	1	in/out in	Port F7: I/O port. CAN receive data.
PG0..PG7 AN0..AN7	8	in in	Port G: Input-only port. Analog input 0 to 7: AD converter input pins.
PL0..PL3 AN8..AN11	4	in in	Port L0 to L3: Input-only port. Analog input 8 to 11: AD converter input pins.
PM0 /$\overline{SS0}$ A16	1	in/out in out	Port M0: I/O port. SEI slave select input 0. Address: Address bus 16.
PM1 MOSI0 A17	1	in/out in/out out	Port M1: I/O port. SEI master output, slave input 0. Address: Address bus 17.

Pin Name	Number of pins	In/Out	Function
PM2 MISO0 A18	1	in/out in/out out	Port M2: I/O port. SEI master input, slave output 0. Address: Address bus 18.
PM3 SECLK0 A19	1	in/out in/out out	Port M3: I/O port. SEI clock input/output 0. Address: Address bus 19.
PM4 $\overline{SS1}$ A20	1	in/out in out	Port M4: I/O port. SEI slave select input. Address: Address bus 20.
PM5 MOSI1 A21	1	in/out in/out out	Port M5: I/O port. SEI master output, slave input 1. Address: Address bus 21.
PM6 MISO1 A22	1	in/out in/out out	Port M6: I/O port. SEI master input, slave output 1. Address: Address bus 22.
PM7 SECLK1 A23	1	in/out in/out out	Port M7: I/O port. SEI clock input/output 1. Address: Address bus 23
PN0 SCK0	1	in/out in/out	Port N0: I/O port. SBI interface 0: clock during SIO mode
PN1 SO0 SDA0	1	in/out out in/out	Port N1: I/O port. SBI interface 0: output data at SIO mode SBI interface 0: data at I²C mode
PN2 SI0 SCL0	1	in/out in in/out	Port N2: I/O port. SBI interface 0: input data at SIO mode SBI interface 0: clock at I²C mode
PN3 SCK1	1	in/out in/out	Port N3: I/O port. SBI interface 1: clock during SIO mode
PN4 SO1 SDA1	1	in/out out in/out	Port N4: I/O port. SBI interface 1: output data at SIO mode SBI interface 1: data at I²C mode
PN5 SI1 SCL1	1	in/out in in/out	Port N5: I/O port. SBI interface 1: input data at SIO mode SBI interface 1: clock at I²C mode
$\overline{NMI}$	1	in	Non-maskable interrupt: Interrupt request pin with programmable falling or both falling and rising edge. 
INT0	1	in	Interrupt request pin 0: Interrupt request pin with programmable level/rising-edge. 
AM0,1	2	in	Address Mode selection: Connect AM0 and AM1 pins to VCC.
TEST0,1	2	in	TEST mode pins :Input "low" when using
CLKOUT1	1	out	Programmable clock output 1
X1/X2	2	in/out	Oscillator connecting pins
$\overline{RESET}$	1	in	Reset input
VREFH	1	in	AD reference voltage high
VREFL	1	in	AD reference voltage low
ADVCC	1	-	Power supply pin for AD converter

Pin Name	Number of pins	In/Out	Function
ADVSS	1	-	GND pin for A/D converter
DVCC5	4	-	Power supply pins (+5V): Conect all DVCC5 pins to 5V power supply.
DVSS5	4	-	GND 5V: Connect all DVSS pins to GND (0V).
DVCC3	4	-	Power supply pins (+3.3V): Connect all DVCC pins to 3.3V power supply.
DVSS3	4	-	GND 3.3V: Connect all DVSS3 pins to GND.(0V)

3. OPERATION

This section describes the basic components, functions and operation of the TMP94FD53F.

3.1 CPU

The TMP94FD53F contains an advanced high-speed 32-bit CPU(900/H2 CPU)

3.1.1 CPU Outline

900/H2 CPU is high-speed and high-performance CPU based on 900/H CPU. 900/H2 CPU has expanded 32-bit internal data bus to process Instructions more quickly.
Outline of 900/H2 CPU are as follows:

	900/H2 CPU	
Width of CPU Address Bus	24-bit	
Width of CPU Data Bus	32-bit	
Internal Operating Frequency	20MHz	
Minimum Bus Cycle	1-clock access(50ns@20MHz)	
Internal RAM	32-bit 1-clock access	
Internal Flash E2PROM	32-bit 1-clock access	
Internal I/O	8/16-bit 2-clock access 8/16-bit 5~6-clock access	900/H2 I/O 900/L1 I/O
External Device	8-bit 2-clock access (can insert some waits)	
Minimum Instruction Execution Cycle	1-clock(50ns@20MHz)	
Conditional Jump	2-clock(100ns@20MHz)	
Instruction Queue Buffer	12-byte	
Instruction Set	Compatible with TLCS-900, 900/L, 900/H, 900/L1 and 900/H2. (NORMAL, MAX, MIN and LDX instruction are deleted)	
CPU mode	No minimum mode	
Micro DMA	8-channel	

3.1.2 Reset operation

When resetting the TMP94FD53 microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overline{\text{RESET}}$ input Low for at least 20 system clocks(4 μ s).At reset the clock doubler is bypassed and system clock operates at 5MHz(fc=20MHz).
When the Reset has been accepted, the CPU performs the following:

- Sets the Program Counter (PC) as follows in accordance with the Reset Vector stored at address FFFF00H~FFFF02H:

  PC<0~7> ← data in location FFFF00H
  PC<8~15> ← data in location FFFF01H
  PC<16~23> ← data in location FFFF02H

- Sets the Stack Pointer (XSP) to 00000000H.
- Sets bits <IFF0~IFF2> of the Status Register (SR) to 111 (thereby setting the Interrupt Level Mask Register to level 7).
- Clears bits <RFP0~RFP1> of the Status Register to 00 (thereby selecting Register Bank 0).

When the Reset is released, the CPU starts executing instructions according to the Program Counter settings. CPU internal registers not mentioned above do not change when the Reset is released.

When the Reset is accepted, the CPU sets internal I/O, ports and other pins as follows.

- Initializes the internal I/O registers as table of "Special Function Register" in Section 5.
- Sets the port pins, including the pins that also act as internal I/O, to General-Purpose Input or Output Port Mode.

Internal reset is released as soon as external reset is released.

The operation of memory controller cannot be insured until power supply becomes stable after power-on reset. The external RAM data provided before turning on the TMP94FD53F may be spoiled because the control signals are unstable until power supply becomes stable after power on reset.

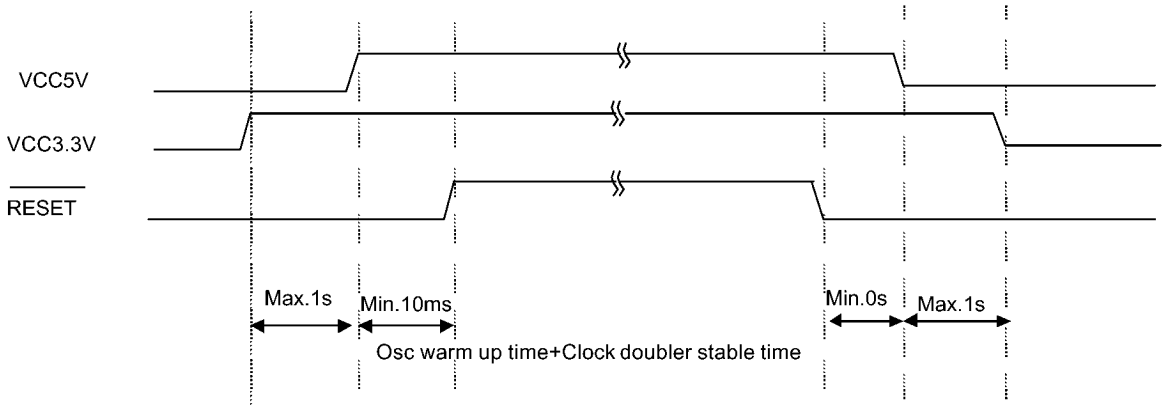


Figure 3.1 Power on Reset Timing Example

3.1.3 Setting of TEST0,TEST1,AM0,AM1

Connect TEST0,TEST1 pin to "GND" to use at NORMAL mode.
Set TEST0 pin to "1" and connect TEST1 pin to "GND" at boot mode for on board programming.
Set AM0,AM1 pin to "1" to use.

Table 3.1.2 Operation Mode Setup Table

Operation Mode	Mode Setup input pin				
	RESET	AM1	AM0	TEST1	TEST0
Single-chip Mode	↑	1	1	0	0
Single-boot Mode		1	1	0	1

3.2 Memory Map

Figure 3.2 is a memory map of the TMP94FD53.

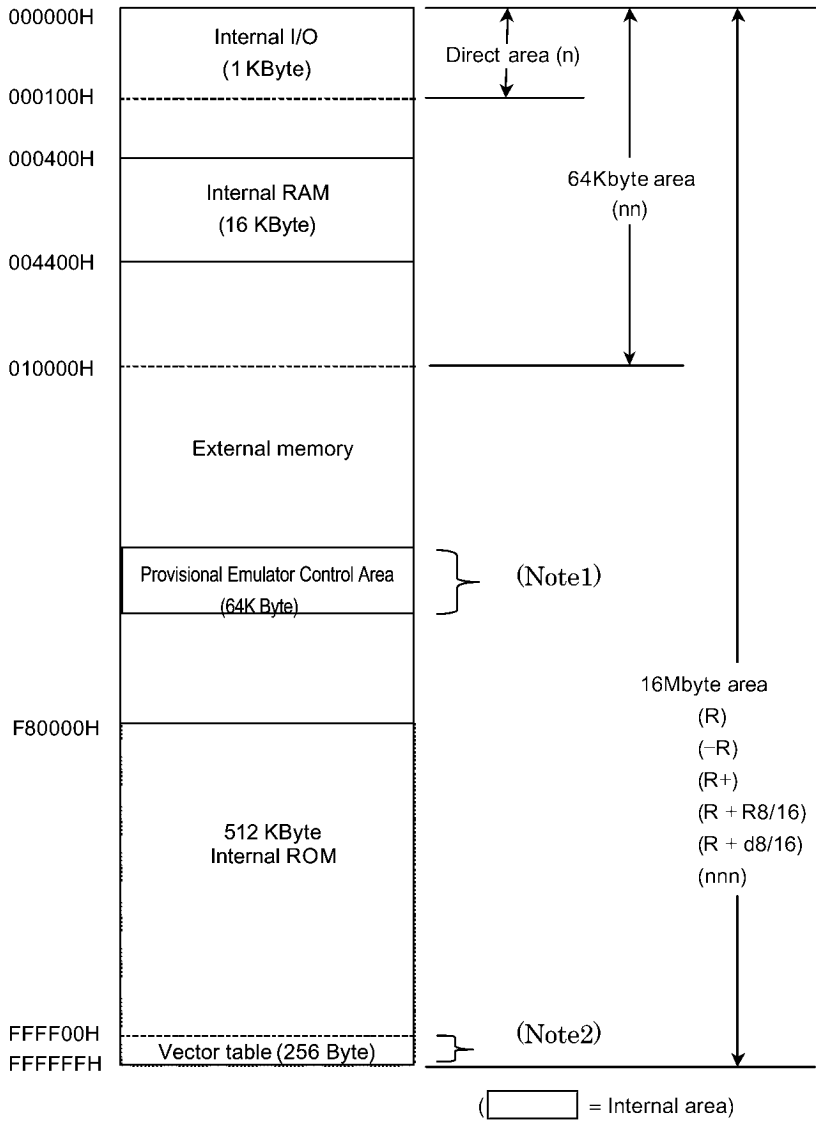


Figure 3.2 Memory Map

Note1: Provisional emulator control area is for emulator, it is mapped F00000H to F10000H address after reset.

Note2: Don't use the last 16-byte area (FFFFF0H to FFFFFFH). This area is reserved.

Note3: On emulator $\overline{WR}$ signal and $\overline{RD}$ signal are asserted, when provisional emulator control area is accessed. Be careful to use external memory.

3.3 Flash Memory

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
  DV_{CC} 3 = 3.3 V ± 0.3 V
  DV_{CC} 5 = 5 V ± 10%
- Functions
  Automatic program
  Automatic erase
  Automatic block erase
  Automatic multiblock erase
- Block erase architecture
  4 Kbytes × 2 / 8 Kbytes × 1/
  16 Kbytes × 1 / 32 Kbytes × 15
- Mode control
  Based on standard JEDEC commands

Block structure:

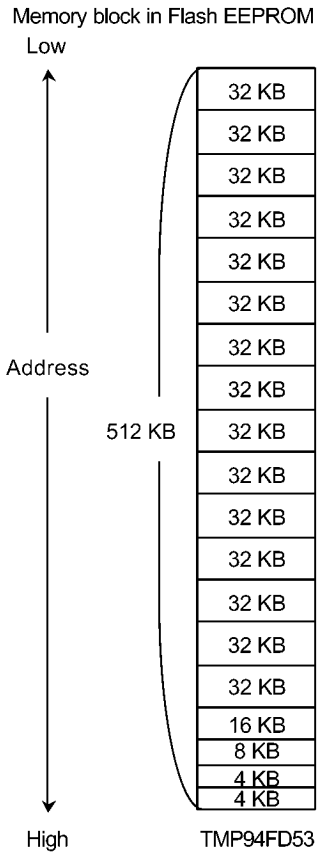


Figure 3.3.1 Block structure of the Flash memory

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

Command Sequence	Bus 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	XXXX0H	F0H	—	—	—	—	—	—	—	—	—	—
Read/ reset	3	XAAA8H	AAH	x5554H	55H	xAAA8H	F0H	RA	RD	—	—	—	—
Auto program	4	XAAA8H	AAH	x5554H	55H	xAAA8H	A0H	PA	PD	—	—	—	—
Auto chip erase	6	XAAA8H	AAH	x5554H	55H	xAAA8H	80H	xAAA8H	AAH	x5554H	55H	xAAA8H	10H
Auto block erase	6	XAAA8H	AAH	x5554H	55H	xAAA8H	80H	xAAA8H	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
xXXX0H	Flash memory address area	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0
xAAA8H		1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	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
PA: Program address
PD: Program data input

Data is written to every quadplex address in units of words.

BA: Block address. Each individual block is selected by a combination of A18, A17, A16, A15, A14, A13 and A12.

*: 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 DQ31 and DQ0 to DQ2 are “Don’t care.”

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

Block	Address in Single Chip Mode							Address Range		Size
	A18	A17	A16	A15	A14	A13	A12	Single Chip	Single Boot	
BA0	L	L	L	L	X	X	X	F80000H-F87FFFH	010000H-017FFFH	32 Kbytes
BA1	L	L	L	H	X	X	X	F88000H-F8FFFFH	018000H-01FFFFH	32 Kbytes
BA2	L	L	H	L	X	X	X	F90000H-F97FFFH	020000H-027FFFH	32 Kbytes
BA3	L	L	H	H	X	X	X	F98000H-F9FFFFH	028000H-02FFFFH	32 Kbytes
BA4	L	H	L	L	X	X	X	FA0000H-FA7FFFH	030000H-037FFFH	32 Kbytes
BA5	L	H	L	H	X	X	X	FA8000H-FAFFFFH	038000H-03FFFFH	32 Kbytes
BA6	L	H	H	L	X	X	X	FB0000H-FB7FFFH	040000H-047FFFH	32 Kbytes
BA7	L	H	H	H	X	X	X	FB8000H-FBFFFFH	048000H-04FFFFH	32 Kbytes
BA8	H	L	L	L	X	X	X	FC0000H-FC7FFFH	050000H-057FFFH	32 Kbytes
BA9	H	L	L	H	X	X	X	FC8000H-FCFFFFH	058000H-05FFFFH	32 Kbytes
BA10	H	L	H	L	X	X	X	FD0000H-FD7FFFH	060000H-067FFFH	32 Kbytes
BA11	H	L	H	H	X	X	X	FD8000H-FDFFFFH	068000H-06FFFFH	32 Kbytes
BA12	H	H	L	L	X	X	X	FE0000H-FE7FFFH	070000H-077FFFH	32 Kbytes
BA13	H	H	L	H	X	X	X	FE8000H-FEFFFFH	078000H-07FFFFH	32 Kbytes
BA14	H	H	H	L	X	X	X	FF0000H-FF7FFFH	080000H-087FFFH	32 Kbytes
BA15	H	H	H	H	L	X	X	FF8000H-FFBFFFH	088000H-08BFFFH	16 Kbytes
BA16	H	H	H	H	H	L	X	FFC000H-FFDFFFH	08C000H-08DFFFH	8 Kbytes
BA17	H	H	H	H	H	H	L	FFE000H-FFEFFFH	08E000H-08EFFFH	4 Kbytes
BA18	H	H	H	H	H	H	H	FFF000H-FFFFFFH	08F000H-08FFFFH	4 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 DQ31 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 quadplex address in units of long 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 (50 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 / $\overline{BY}$ (Ready / Busy)

This function is not supported.

(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."

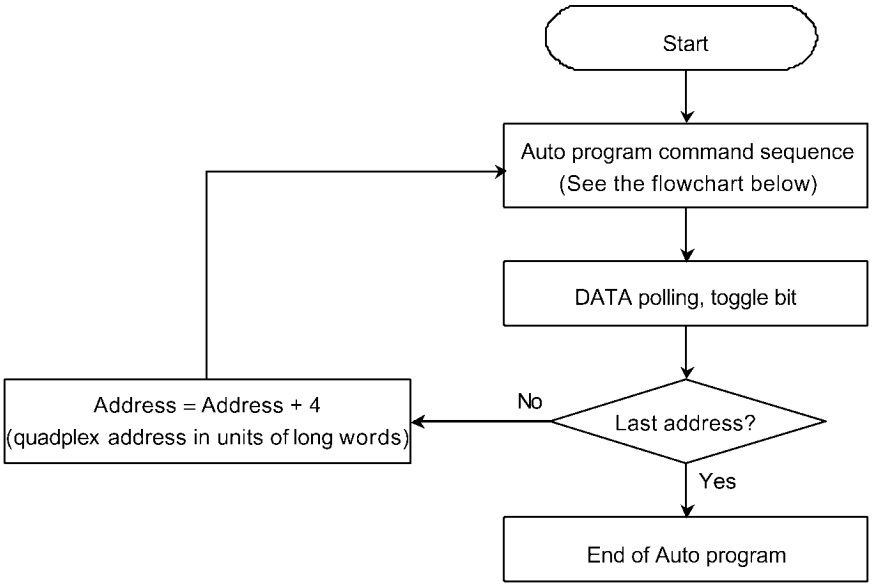
2) User boot

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.

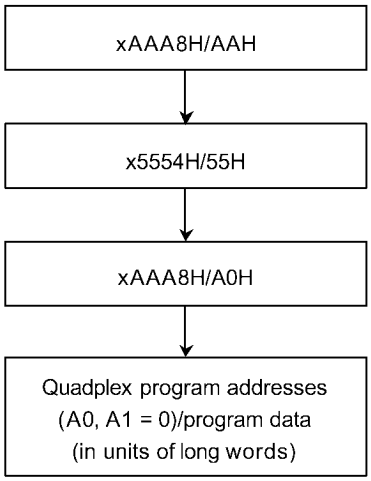
The flash memory rewrite program including routines for taking in the rewrite data and rewriting the flash memory needs to be prepared in advance. When in the main program, switch from regular operation to the flash memory rewrite operation, then execute the flash memory rewrite program you've prepared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

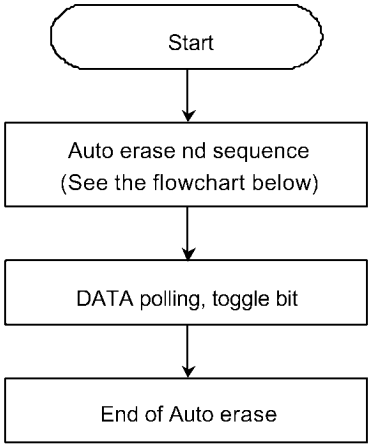
Auto program



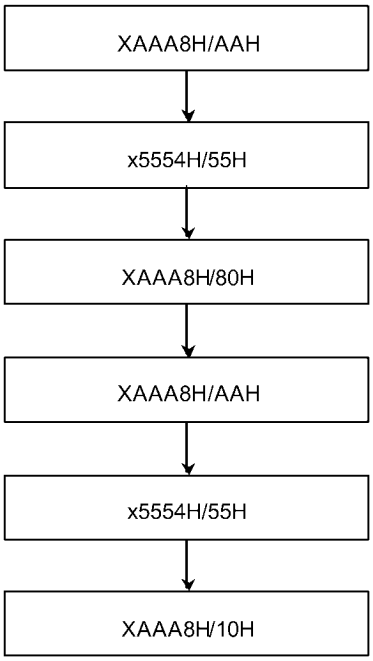
Auto program command sequence (addresses/commands)



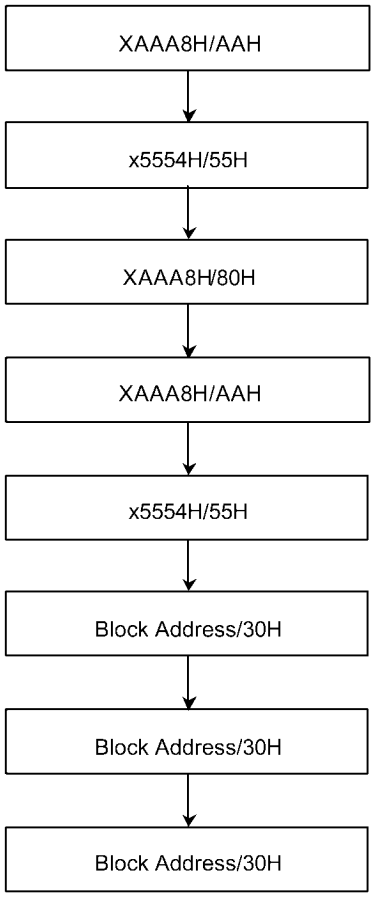
Auto erase



Auto chip erase command sequence
(addresses/commands)

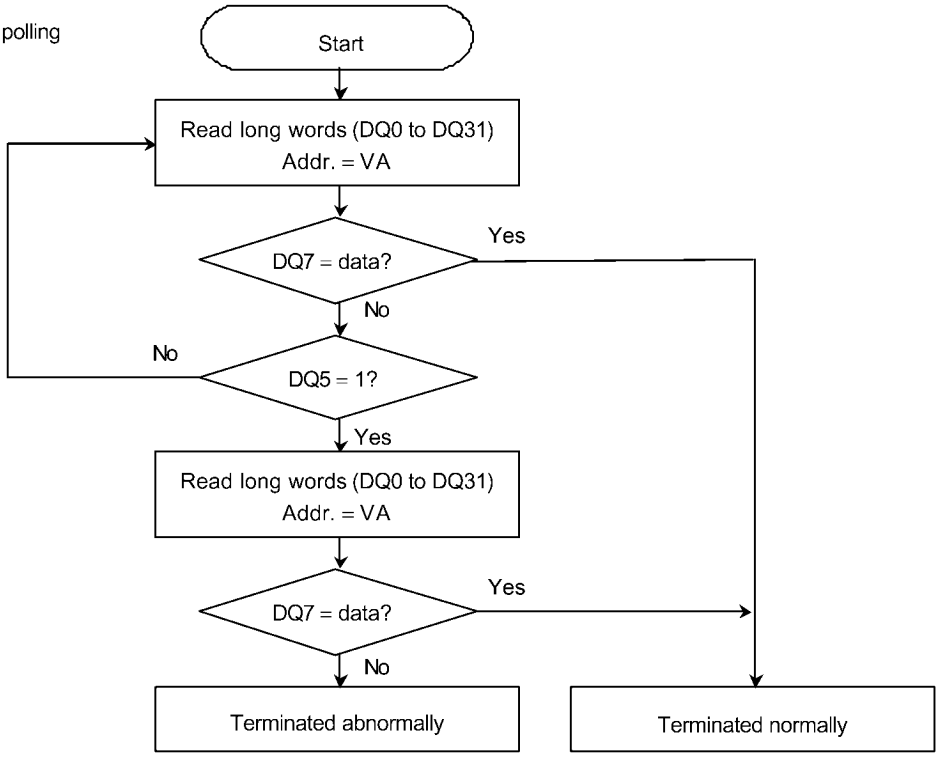


Auto block/multiblock erase command sequence
(addresses/commands)

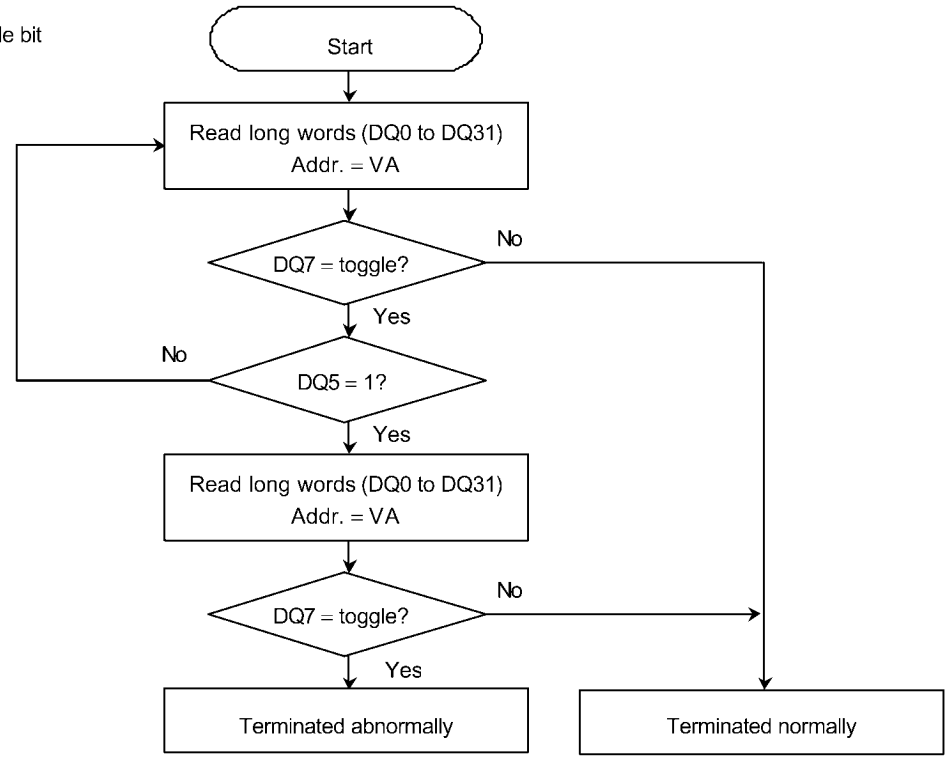


Assitional address input during Auto multiblock erase (each within 50μs)

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 TMP94FD53 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 3.5.1 shows an example of how to connect the programming controller and the target board.

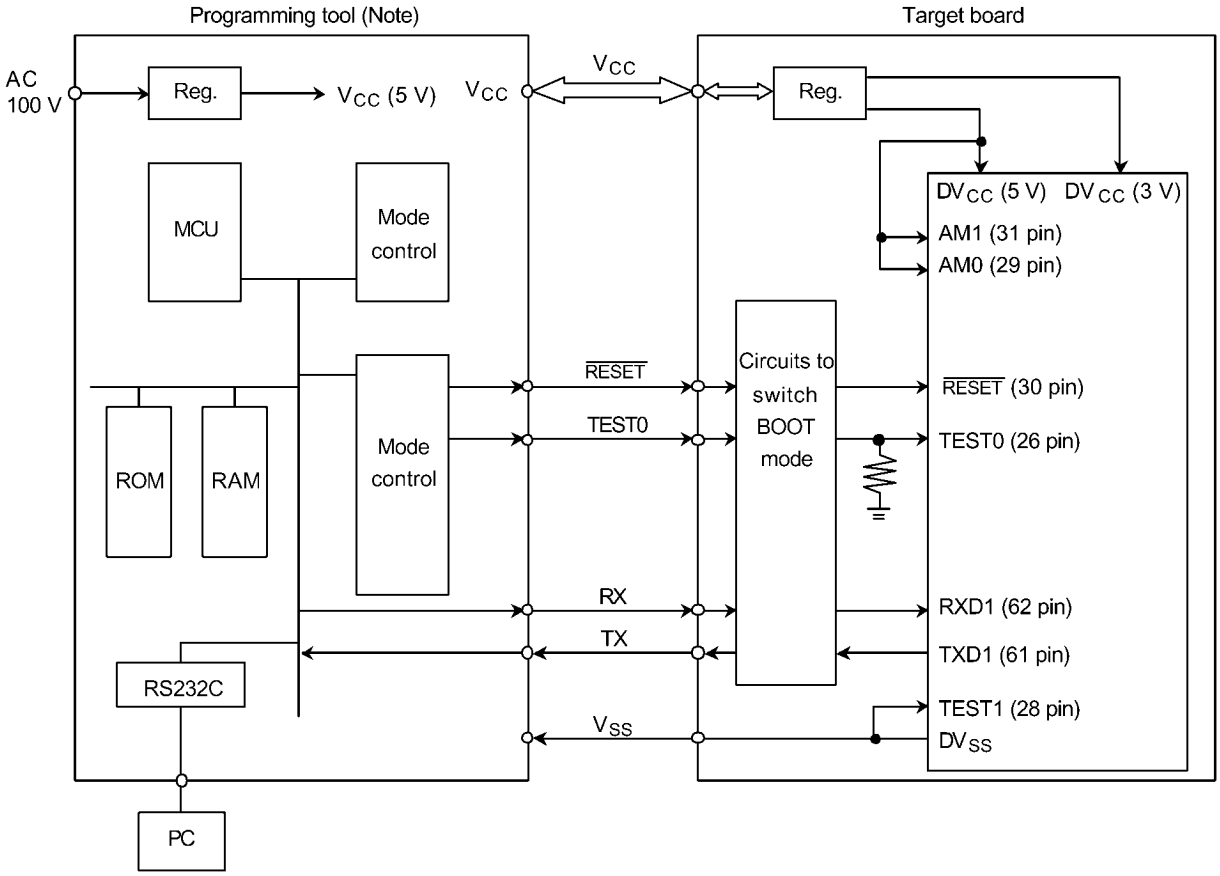


Figure 3.5.1 Example for Connecting Units for On-board Programming

Note: One of the programming controllers supported for the TMP94FD53 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:

North America

Yokogawa Digital Computer America

Tel : 408-244-1932 Fax : 408-244-1881

European Area

Ashling Microsystems Limited

Tel : 353-61-334466 Fax : 353-61-334477
Tel : 44-1256-811998 Fax : 44-1256-811761

Korea

KM DATA INC.

Tel : 82-2-785-3929 Fax : 82-2-785-3117

(2) Mode settings

To execute on-board programming, start the TMP94FD53 in single-boot mode. Settings necessary to start up in single-boot mode are shown below.

TEST0 = H
TEST1 = L
AM0 = H
AM1 = H
$\overline{\text{RESET}}$ = ↗

After setting the TEST0, TEST1, AM1, and AM0 pins each to the above conditions, drive the signal input to the $\overline{\text{RESET}}$ pin high. The TMP94FD53 starts up in single-boot mode.

(3) Memory map

Figure 3.5.2 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 8FFFFH, as shown here.

You'll also find that the boot ROM (MROM) is mapped into addresses FFF400H through FFFFFFH.

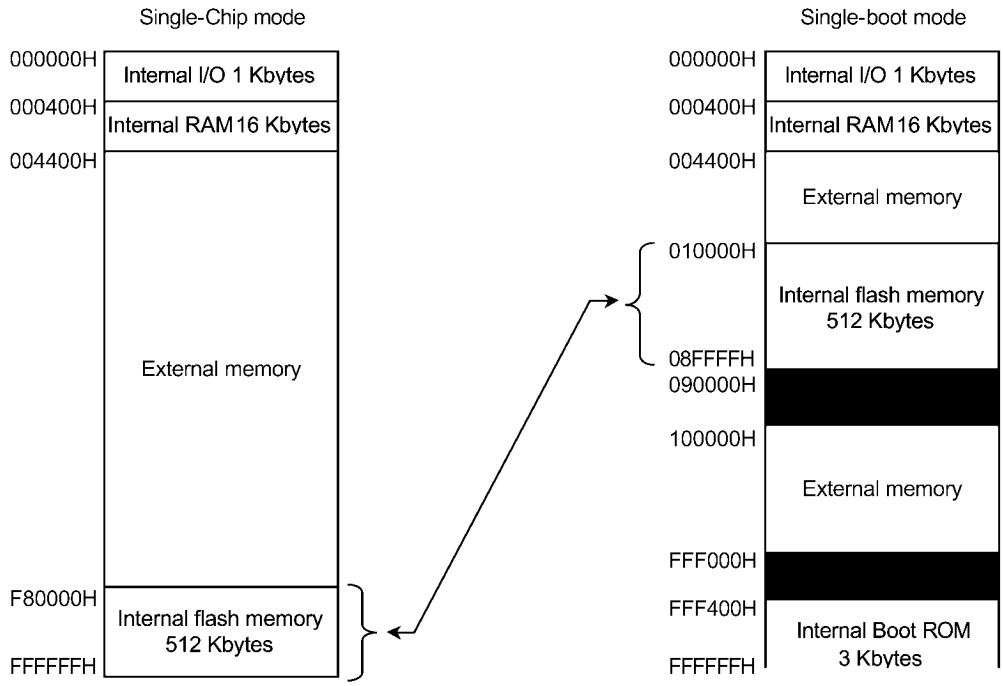


Figure 3.5.2 Comparison of Memory Maps

(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 TMP94FD53.

Note that although the default baud rate is 9,600 bps (@ fc = 20 MHz), it can be changed to other values as shown in Table 3.5.1.

Communication channel: SIO channel 1

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

Data length: 8 bits

Parity it: None

Stop bit: 1 bit

Baud rate (default): 9,600 bps (@ fc = 20 MHz)

(5) Data transfer format

Tables 3.5.1 through 3.5.6 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 3.5.1 Baud Rate Modification Data (@ fc = 20 MHz)

Baud Rate Modification Data	04H	05H	07H	0AH	18H	28H
Baud Rate (bps)	76800	62500	38400	31250	19200	9600

Table 3.5.2 Operation Command Data

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

Table 3.5.3 Boot Program Transfer Format (@fc = 20 MHz) (For Flash Memory Rewrite)

	Number of Bytes Transferred	Transfer Data from Controller to TMP94FD53	Baud Rate	Transfer Data from TMP94FD53 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 3.5.1) —	9600 bps 9600 bps	— OK: Echoback data NG: A1H × 3, A2H × 3, A3H × 3 63H × 3
	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	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: "xxH×3" 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 3.5.4 Boot Program Transfer Format (@fc = 20 MHz) (For RAM Loader)

	Number of Bytes Transferred	Transfer Data from Controller to TMP94FD53	Baud Rate	Transfer Data from TMP94FD53 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 3.5.1) —	9600 bps 9600 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: "xxH×3" 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 3.5.5 Boot Program Transfer Format (@fc = 20 MHz) (For Flash Memory SUM)

	Number of Bytes Transferred	Transfer Data from Controller to TMP94FD53	Baud Rate	Transfer Data from TMP94FD53 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 3.5.1) —	9600 bps 9600 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 : n'th – 2 byte	—	Changed new baud rate	OK: SUM (Low) NG: —
	9th byte	(Wait for the next operation command data)	Changed new baud rate	—

*1: "xxH×3" 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 TMP94FD53 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.

- Flash memory rewrite

  The flash memory is erased the entire chip (512 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 512 Kbytes of flash memory is calculated and the result is returned to the controller.

- 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.

- Flash memory SUM

  The SUM of 512 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 3.5.3)

- 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 9,600 bps.
  The matching data is 5AH.
- 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.
- The receive data in the 3rd byte is the baud rate modification data. The seven kinds of baud rate modification data shown in Table 3.5.1 are available. Even when you do not change the baud rate, be sure to send the initial baud rate data (28h: 9,600 bps @fc = 20 MHz).
  Baud rate modification becomes effective after the echoback transmission is completed.
- 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).
- The receive data in the 5th byte is the command data (30H) to rewrite the flash memory.
- 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 3.5.2. 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).
- The transmit data in the 7th byte indicates whether collective erase (512 Kbytes) has terminated normally. When collective erase (512 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).
- 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.

- 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 512 Kbytes of flash memory area is approximately 800 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.
- 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 3.5.4)

- The transmit/receive data in the 1st through the 4th bytes are the same as in the case of flash memory rewrite commands.
- The receive data in the 5th byte is the RAM loader command data (60H).
- 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 3.5.2. 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.

- 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.
- 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.
- 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 3.5.5)

- The transmit/receive data in the 1st through the 4th bytes are the same as in the case of flash memory rewrite commands.
- The receive data in the 5th byte is the flash memory SUM command data (90H).
- 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 3.5.2. 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.
- 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.
- 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.

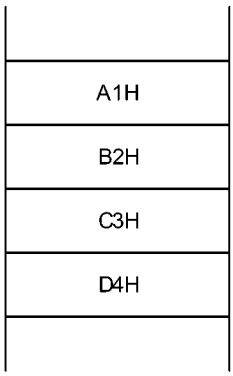
Table 3.5.6 Boot Program Transmit Data

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

- 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.



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.

- Calculation data
  The data from which SUM is calculated are listed in Table 3.5.7 below.

Table 3.5.7 SUM Calculation Data

Operation Mode	Calculation Data	Remarks
Flash memory rewrite command	Data in the entire area (512 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 (512 Kbytes) of flash memory	—

⑥ Notes on Extended Intel Hex Format (binary)

- 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 TMP 94FD53 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.
- 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.
- 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.
- 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.
- 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 3.5.8.

Table 3.5.8 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 3.5.3)	Data
Controller to TMP94FD53	Extended record	: 02 0000 02 1000 EC zz
Controller to TMP94FD53	Data record (data length: 08H)	: 08 FFF8 00 xxxxxx CS zz
Controller to TMP94FD53	Extended record	: 02 0000 02 2000 DC zz
Controller to TMP94FD53	Data record (data length: 30H)	: 30 0000 00 yyyyyyyy CS zz
Controller to TMP94FD53	End record	: 00 0000 01 FF ww
TMP94FD53 to controller	SUM (upper byte) (n'th − 1 byte in Table 3.5.3)	SUM (upper byte)
TMP94FD53 to controller	SUM (lower byte) (n'th byte in Table 3.5.3)	SUM (lower byte)
Controller to TMP94FD53	Operation command (n'th + 1 byte in Table 3.5.3)	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 72000H to 8DFFFH. Figure 3.5.3 schematically shows the password area.

- 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 72000H

  Address 8DFFFH < PNSA

  N < 8

- 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 72000H

  Address 8DFFFH < 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 (8FF00H to 8FFFFH) are FFH, the device is assumed to be a blank product, in which no check is made of the passwords.

- 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

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

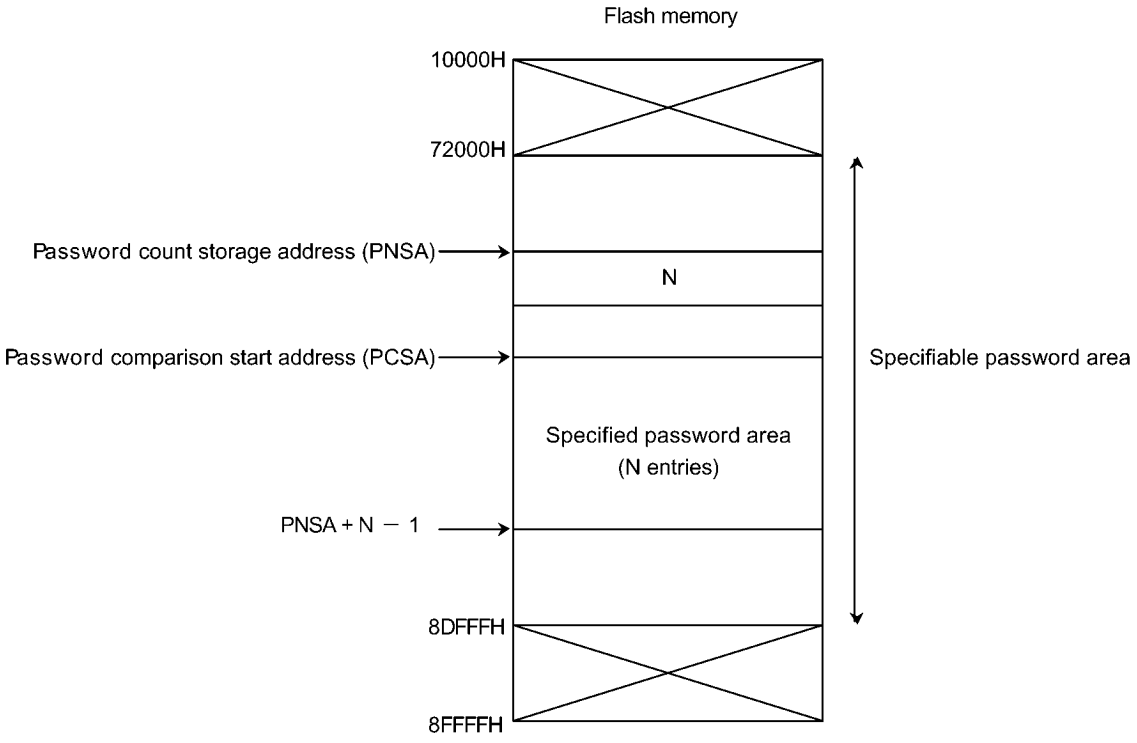
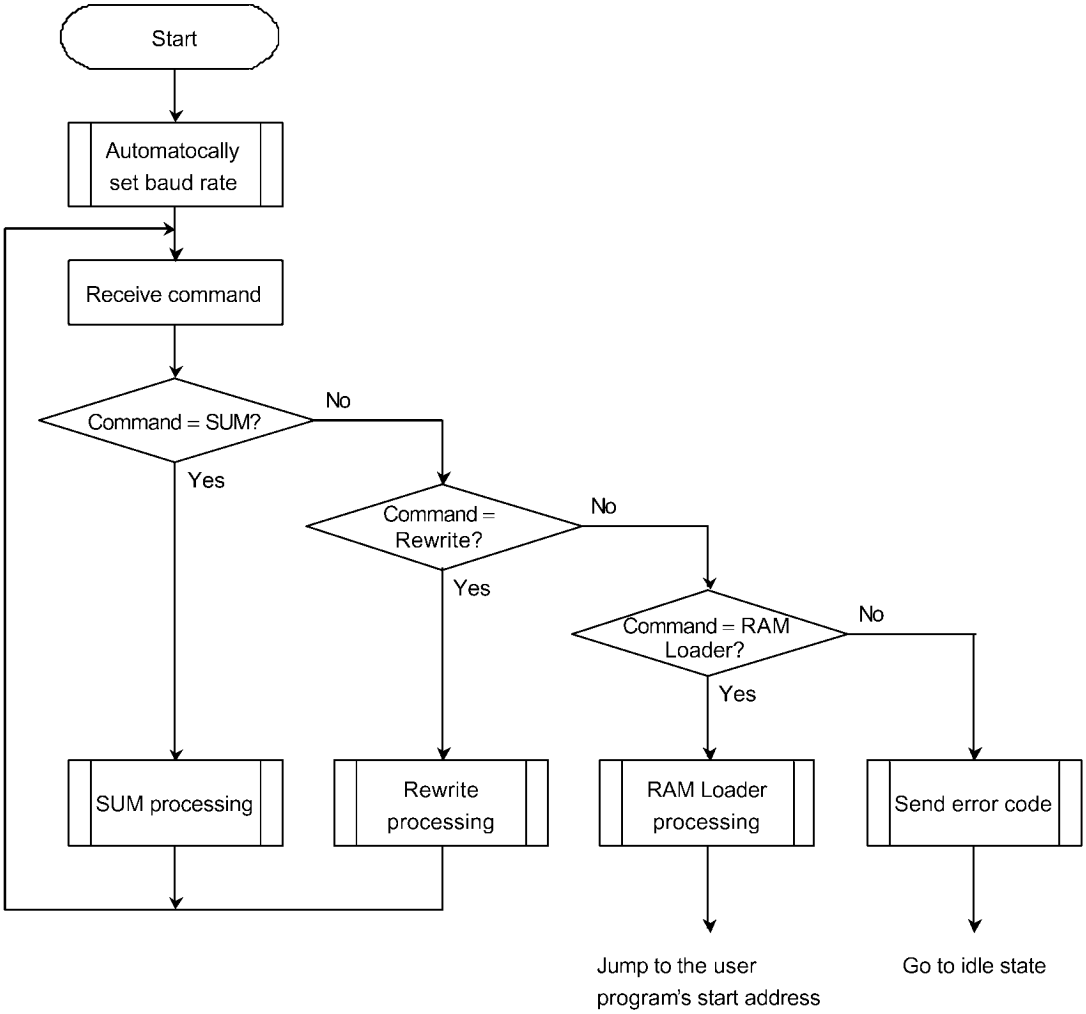
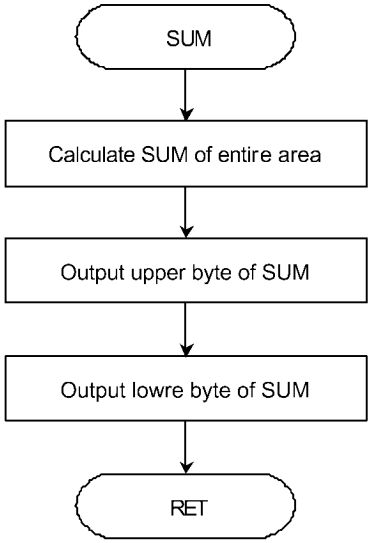


Figure 3.5.3 Conceptual Diagram of a Password Area

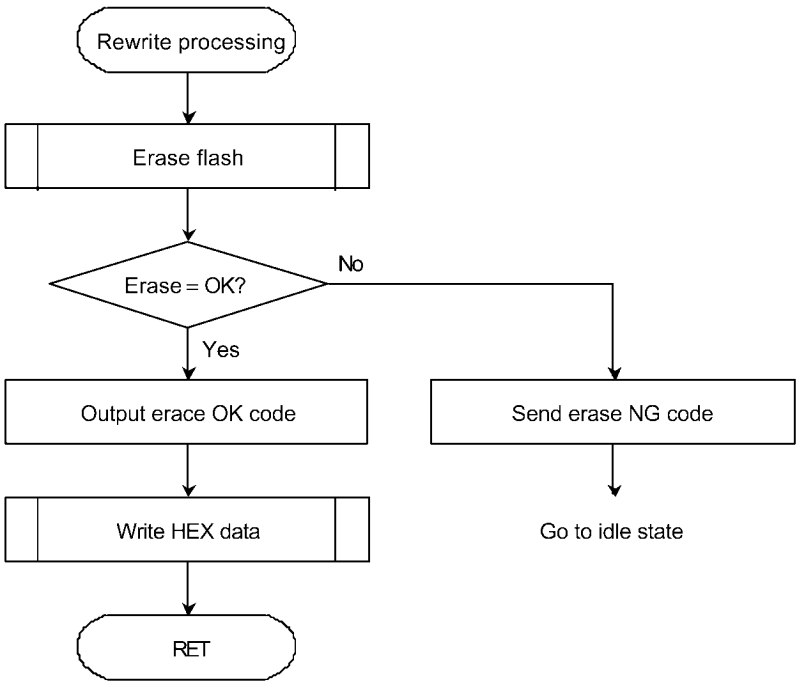
Single Boot General Flow



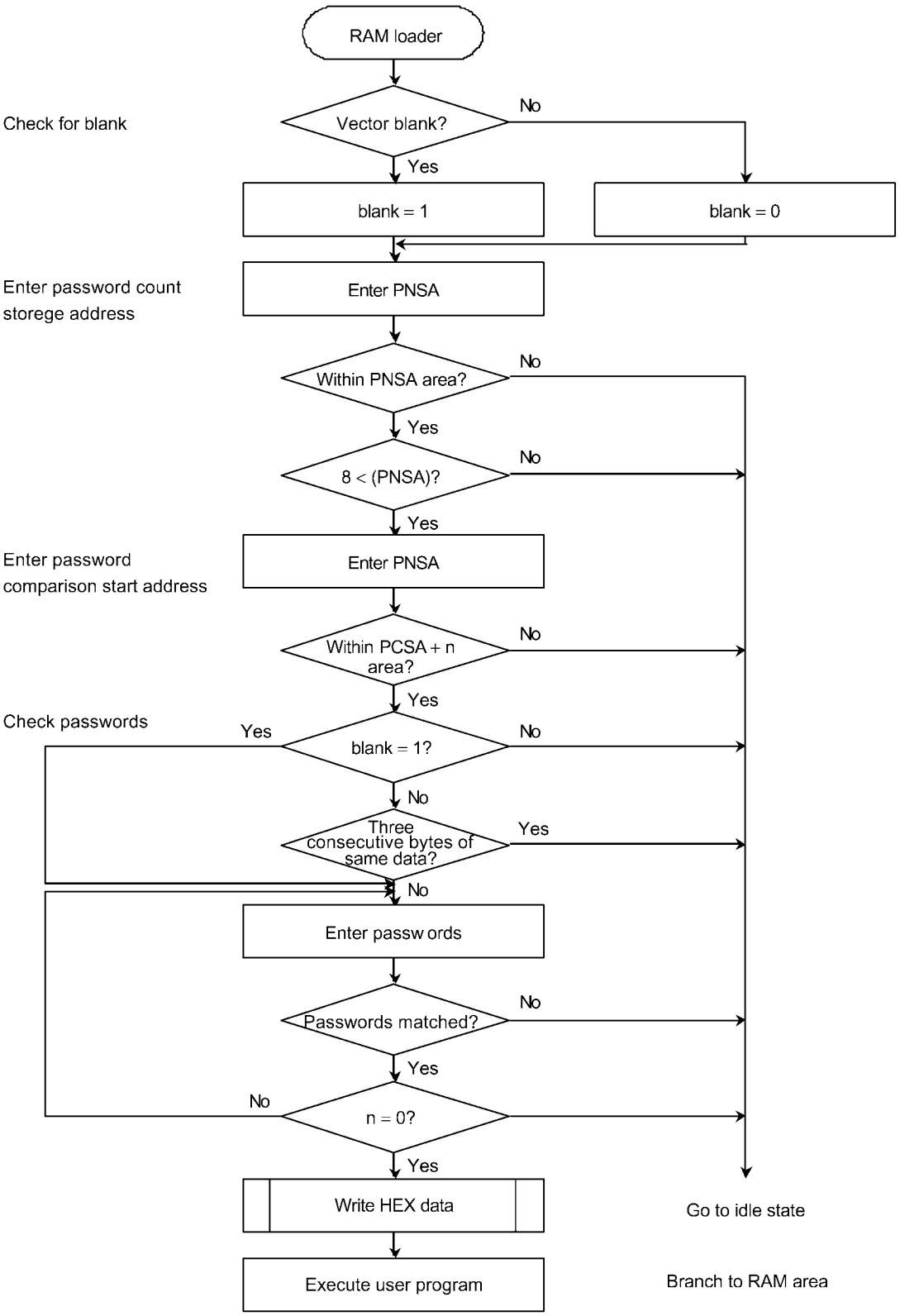
(1) SUM command



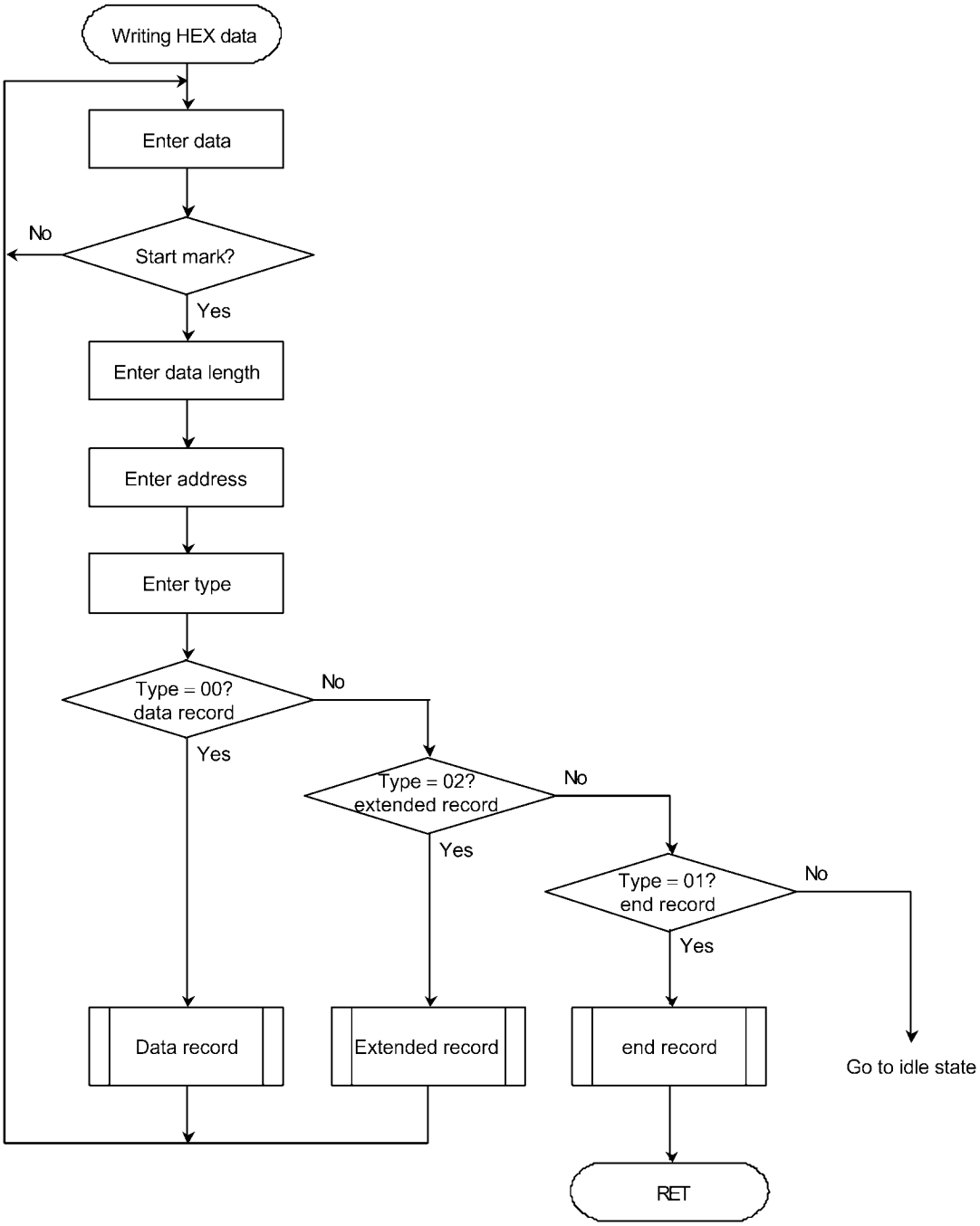
(2) Rewrite command



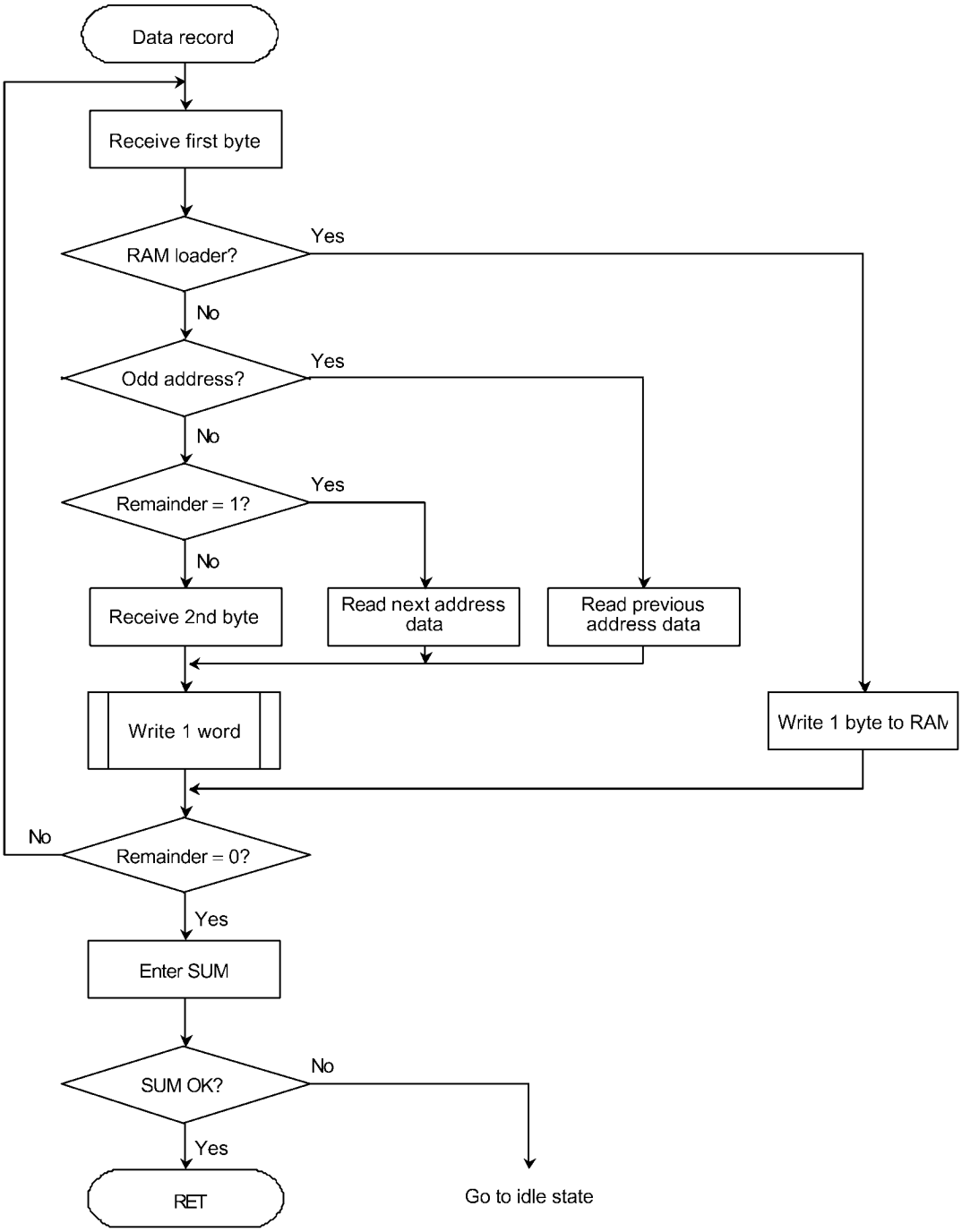
(3) RAM loader command



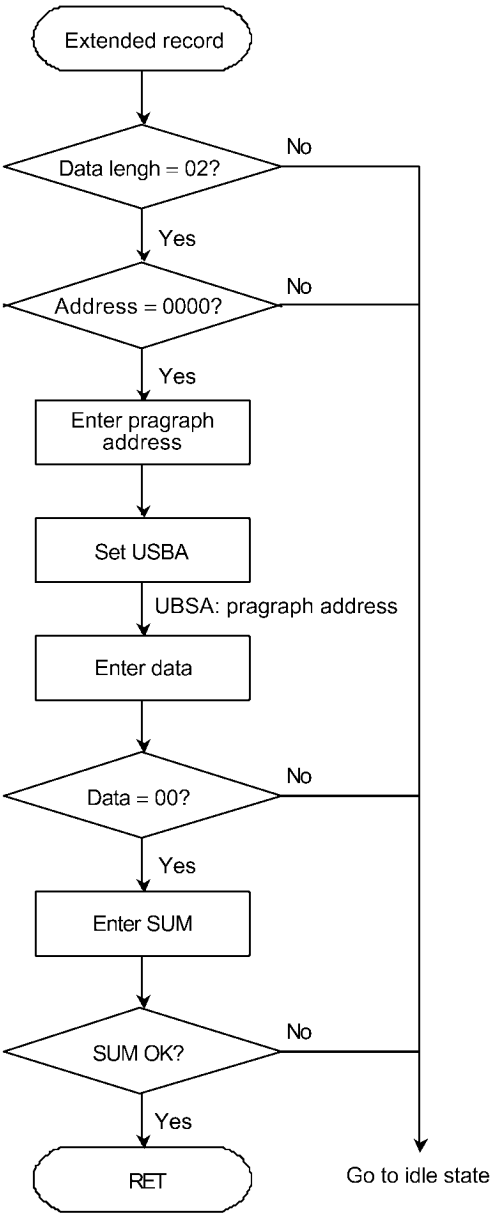
(2) -1 Writing HEX data



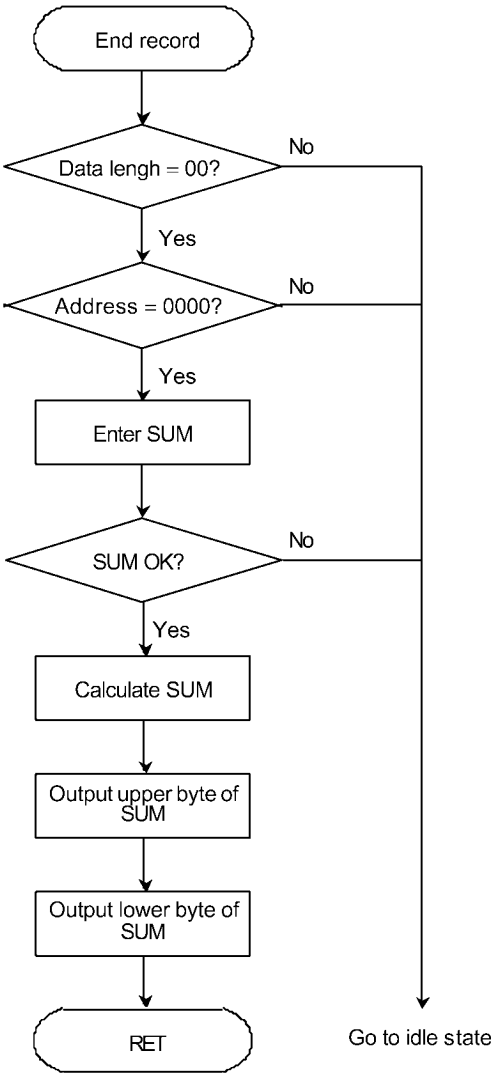
(2) -1-1 Data record



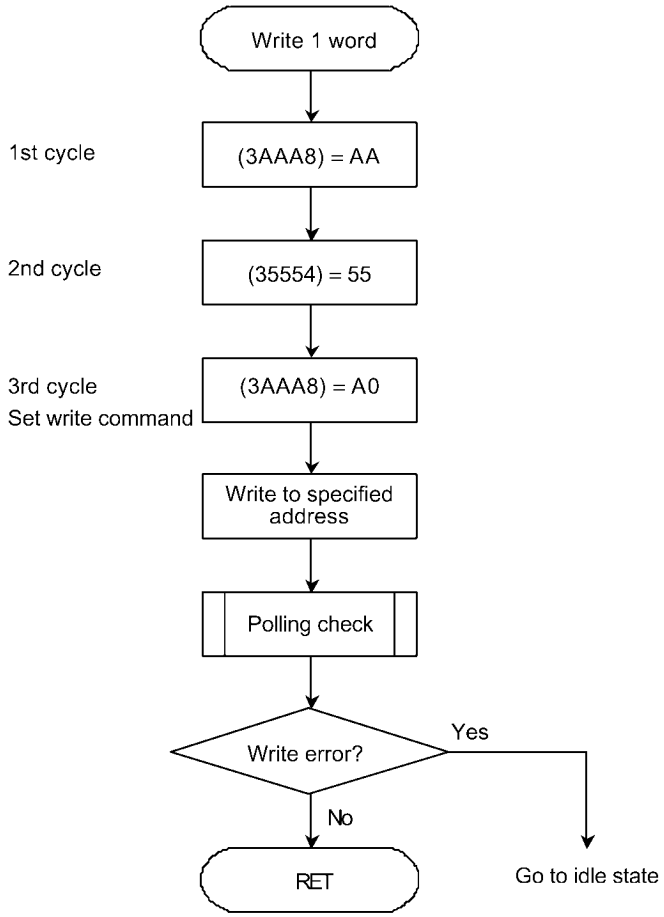
(2) -1-2 Extended record



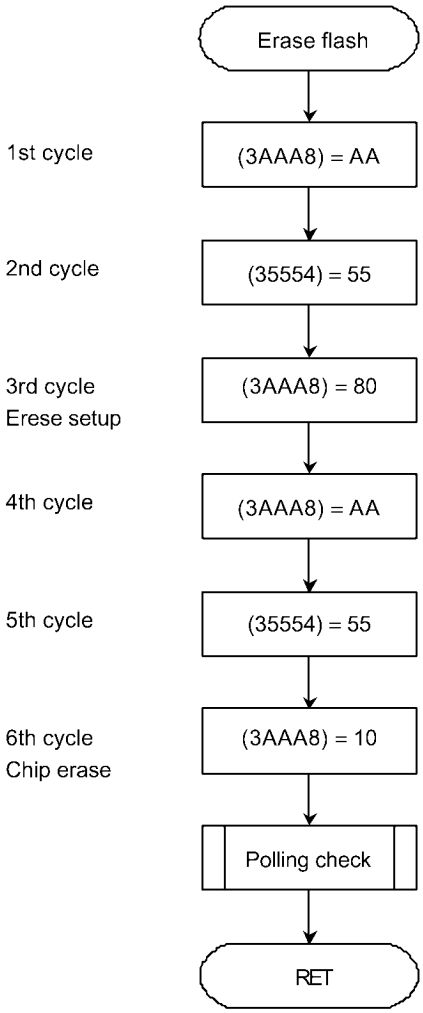
(2) -1-3 End record



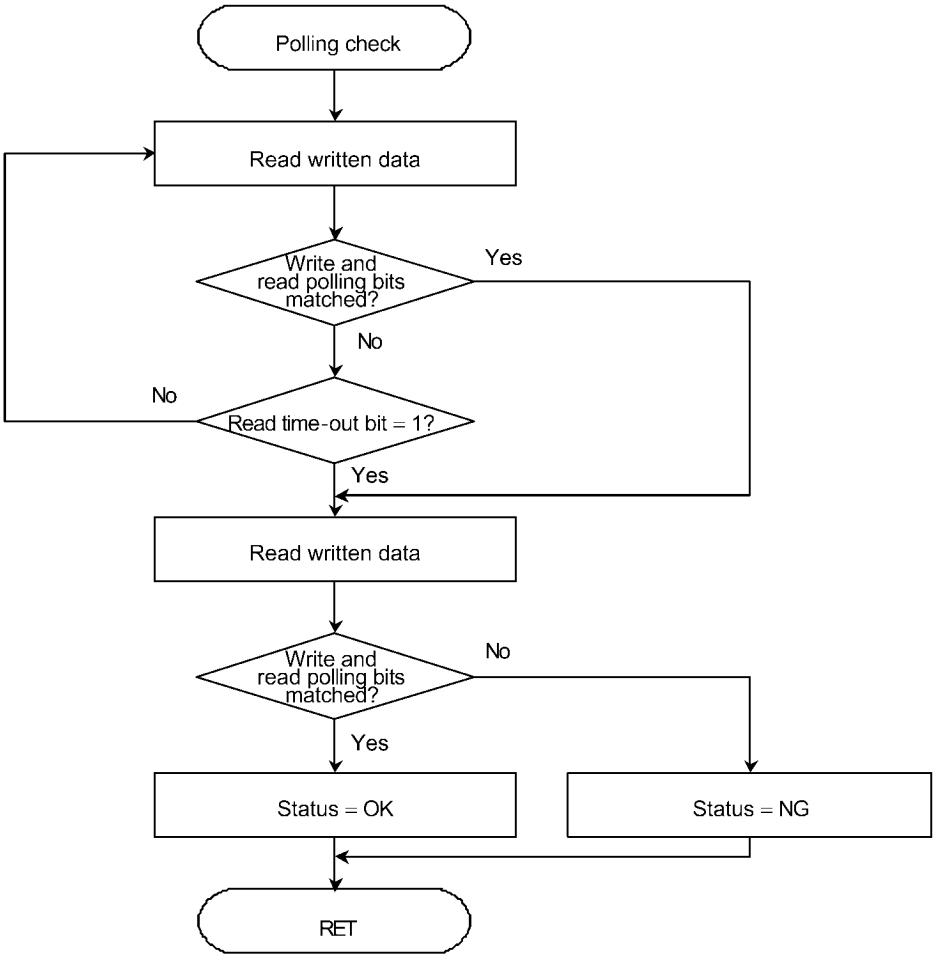
(2) -1-1-1 Writing one word



(2) –2 Flash mempry erase



(2) -2-1 Data polling



4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power Supply Voltage	V_{CC3}	-0.5 to 4.5	V
	V_{CC5}	-0.5 to 6.5	
Input Voltage	V_{IN}	-0.5 to VCC3+0.5	V
		-0.5 to VCC5+0.5	
Output Current (total)	ΣI_{OL}	100	mA
Output Current (total)	ΣI_{OH}	-100	mA
Power Dissipation (Ta=85°C)	P_D	600	mW
Soldering Temperature (10s)	T_{SOLDER}	260	°C
Storage Temperature	T_{STG}	-65 to 150	°C
Operation Temperature	T_{OPR}	-40 to 85	°C
Operation Temperature (FLASH Program/Erase)		0 to 70	
Number of Times Program Erased	N_{EW}	100	Cycle

Note: The absolute maximum ratings are rated values that 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 that include this device, ensure that no absolute maximum rating value will ever be exceeded.

4.2 DC Electrical Characteristics

Vcc3 = 3.0 to 3.6V / Vcc5 =4.5V to 5.5V / fc = 16 to 20MHz / Ta = -40 to 85℃

Parameter	Symbol	Condition	Min	Max	Unit
Supply Voltage	V_{CC3}		3.0	3.6	V
	V_{CC5}		4.5	5.5	
Input Low Voltage P00 to P07 (D0 to 7) PG0 to PG7 PL0 to PL3	V_{IL0}		-0.3	0.8	V
Input Low Voltage P00 to P07 (PORT) P40 to P47 P70 to P75 PC0 to PC5 PD0 to PD7 PF0 to PF7 PM0 to PM7 PN0 to PN5	V_{IL1}		-0.3	0.3*VCC5	V
Input Low Voltage INT0 $\overline{NMI}$ $\overline{RESET}$	V_{IL2}		-0.3	0.25*VCC5	V
Input Low Voltage AM0 to AM1 TEST0 to TEST1	V_{IL3}		-0.3	0.3	V
Input Low Voltage X1	V_{IL4}		-0.3	0.2*VCC3	V
Input High Voltage P00 to P07 (D0 to 7) PG0 to PG7 PL0 to PL3	V_{IH0}		2.2	VCC5+0.3	V
Input High Voltage P00 to P07 P40 to P47 P70 to P75 PC0 to PC5 PD0 to PD7 PF0 to PF7 PM0 to PM7 PN0 to PN5	V_{IH1}		0.7*VCC5	VCC5+0.3	V
Input High Voltage INT0 $\overline{NMI}$ $\overline{RESET}$	V_{IH2}		0.75*VCC5	VCC5+0.3	V
Input High Voltage AM0 to AM1 TEST0 to TEST1	V_{IH3}		VCC5-0.3	VCC5+0.3	V
Input High Voltage X1	V_{IH4}		0.8*VCC3	VCC3+0.3	V

Parameter	Symbol	Condition		Min	Max	Unit
Output Low Voltage	V_{OL}	I_{OL} = 1.6mA			0.45	V
Output High Voltage	V_{OH0}	I_{OH} = -400μA		2.4		V
	V_{OH1}	I_{OH} = -100μA		0.75*VCC5		
	V_{OH2}	I_{OH} = -20μA		0.9*VCC5		
Input Leakage Current	I_{LI}	0.0 ≦ Vin ≦ VCC5		0.02 (typ.)	±5	μA
	I_{L2}	0.0 ≦ Vin ≦ VCC5 (PortG, PortL)		0.02 (typ.)	±0.5	
Output Leakage Current	I_{LO}	0.2 ≦ Vin ≦ VCC5-0.2		0.05 (typ.)	±10	μA
Operating Current (Single Chip)*	I_{CC3}	V_{CC3}=3.3V , X1=10MHz (Internal 20MHz) (Includes I_{CCPLL})		85 (typ)	100	mA
	I_{CC5}	V_{CC5}=5.0V , X1=10MHz (Internal 20MHz)		0.5 (typ)	30	
Operating Current (Stand-by)	$I_{CC3IDLE2}$	IDLE2 Mode	V_{CC3}=3.6V, X1=10MHz (Internal 20MHz)		90	mA
	$I_{CC5IDLE2}$		V_{CC5}=5.5V, X1=10MHz (Internal 20MHz)		10	
	$I_{CC3IDLE1}$	IDLE1 Mode	V_{CC3}=3.6V, X1=10MHz (Internal 20MHz)		20	
	$I_{CC5IDLE1}$		V_{CC5}=5.5V, X1=10MHz (Internal 20MHz)		10	
	$I_{CC3STOP}$	STOP Mode	V_{CC3}=3.6V		300	μA
	$I_{CC5STOP}$		V_{CC5}=5.5V		300	
Stand-by Voltage	V_{STB3}	V_{DD3} < V_{DD5} , V_{IH1}<V_{DD5} , V_{IH2}<V_{DD5} , V_{IH3}<V_{DD5}		2.5	3.6	V
	V_{STB5}			2.5	5.5	
Pull-up Resistor	R_{RST}	$\overline{RESET}$		60	220	KΩ
	R_{CLK}	CLK				
Pin Capacitance	C_{IO}	fc=1MHz			10	pF
Schmitt Width	V_{TH}	INT0, $\overline{NMI}$, $\overline{RESET}$		0.4	1.0 (typ.)	V

*: On condition that external bus don't operate

4.3 AC Characteristics

Read cycle

VCC3=3.3V±0.3V, VCC5=5.0V±10%, TA=-40 to 85℃

No.	Parameter	Symbol	Min	Max	@20MHz	@16MHz	Unit
1	OSC period (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System Clock period (=T)	t_{CYC}	50	62.5	50	62.5	ns
3	CLKOUT1 Low Width	t_{CL}	0.5×T-15		10	16	ns
4	CLKOUT1 High Width	t_{CH}	0.5×T-15		10	16	ns
5-1	A0 to A23 Valid → D0 to D7 Input @0WAIT	t_{AD}		2.0×T-50	50	75	ns
5-2	A0 to A23 Valid → D0 to D7 Input @1WAIT	t_{AD3}		3.0×T-50	100	138	ns
6-1	$\overline{RD}$ Fall → D0 to D7 Input @0WAIT	t_{RD}		1.5×T-45	30	49	ns
6-2	$\overline{RD}$ Fall → D0 to D7 Input @1WAIT	t_{RD3}		2.5×T-45	80	111	ns
7-1	$\overline{RD}$ Low Width @0WAIT	t_{RR}	1.5×T-20		55	74	ns
7-2	$\overline{RD}$ Low Width @1WAIT	t_{RR3}	2.5×T-20		105	136	ns
8	A0 to A23 Valid → $\overline{RD}$ Fall	t_{AR}	0.5×T-20		5	11	ns
9	$\overline{RD}$ Fall → CLK Fall	t_{RK}	0.5×T-20		5	11	ns
10	A0 to A23 Valid → D0 to D7 Hold	t_{HA}	0		0	0	ns
11	$\overline{RD}$ Rise → D0 to D7 Hold	t_{HR}	0		0	0	ns
12	A0 to A23 Valid → PORT Input	t_{APR}		2.0×T-120	-20	5	ns
13	A0 to A23 Valid → PORT Hold	t_{APH}	2.0×T		100	125	ns
14	$\overline{WAIT}$ Set-up Time	t_{TK}	15		15	15	ns
15	$\overline{WAIT}$ Hold Time	t_{KT}	5		5	5	ns

Write cycle

VCC3=3.3V±0.3V, VCC5=5.0V±10%, TA=-40 to 85℃

No.	Parameter	Symbol	Min	Max	@20MHz	@16MHz	Unit
1	OSC period (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System Clock period (=T)	t_{CYC}	50	62.5	50	62.5	ns
3	CLKOUT1 Low Width	t_{CL}	0.5×T-15		10	16	ns
4	CLKOUT1 High Width	t_{CH}	0.5×T-15		10	16	ns
5-1	D0 to D7 Valid → $\overline{WR}$ Rise @0WAIT	t_{DW}	1.25×T-35		28	43	ns
5-2	D0 to D7 Valid → $\overline{WR}$ Rise @1WAIT	t_{DW3}	2.25×T-35		78	106	ns
6-1	$\overline{WR}$ Low Width @0WAIT	t_{WW}	1.25×T-30		33	48	ns
6-2	$\overline{WR}$ Low Width @1WAIT	t_{WW3}	2.25×T-30		83	111	ns
7	A0 to A23 Valid → $\overline{WR}$ Fall	t_{AW}	0.5×T-20		5	11	ns
8	$\overline{WR}$ Fall → CLK Fall	t_{WK}	0.5×T-20		5	11	ns
9	$\overline{WR}$ Fall → A0 to A23 Hold	t_{WA}	0.25×T-5		8	11	ns
10	$\overline{WR}$ Fall → D0 to D7 Hold	t_{WD}	0.25×T-5		8	11	ns
11	A0 to A23 Valid → PORT Output	t_{APW}		2.0×T+70	170	195	ns
12	$\overline{WAIT}$ Set-up Time	t_{TK}	15		15	15	ns
13	$\overline{WAIT}$ Hold Time	t_{KT}	5		5	5	ns
14	$\overline{RD}$ Rise → D0 to D7 Output	t_{RDO}	1.25×T-35		20	26	ns

AC Condition

●Output : P0(D0 to D7), P4(A0 to A7), PD(A8 to A15), PM(A16 to A23), P70(RD), P71(WR)

High 2.0V, Low 0.8V, CL=50pF

Others

High 2.0V, Low 0.8V, CL=50pF

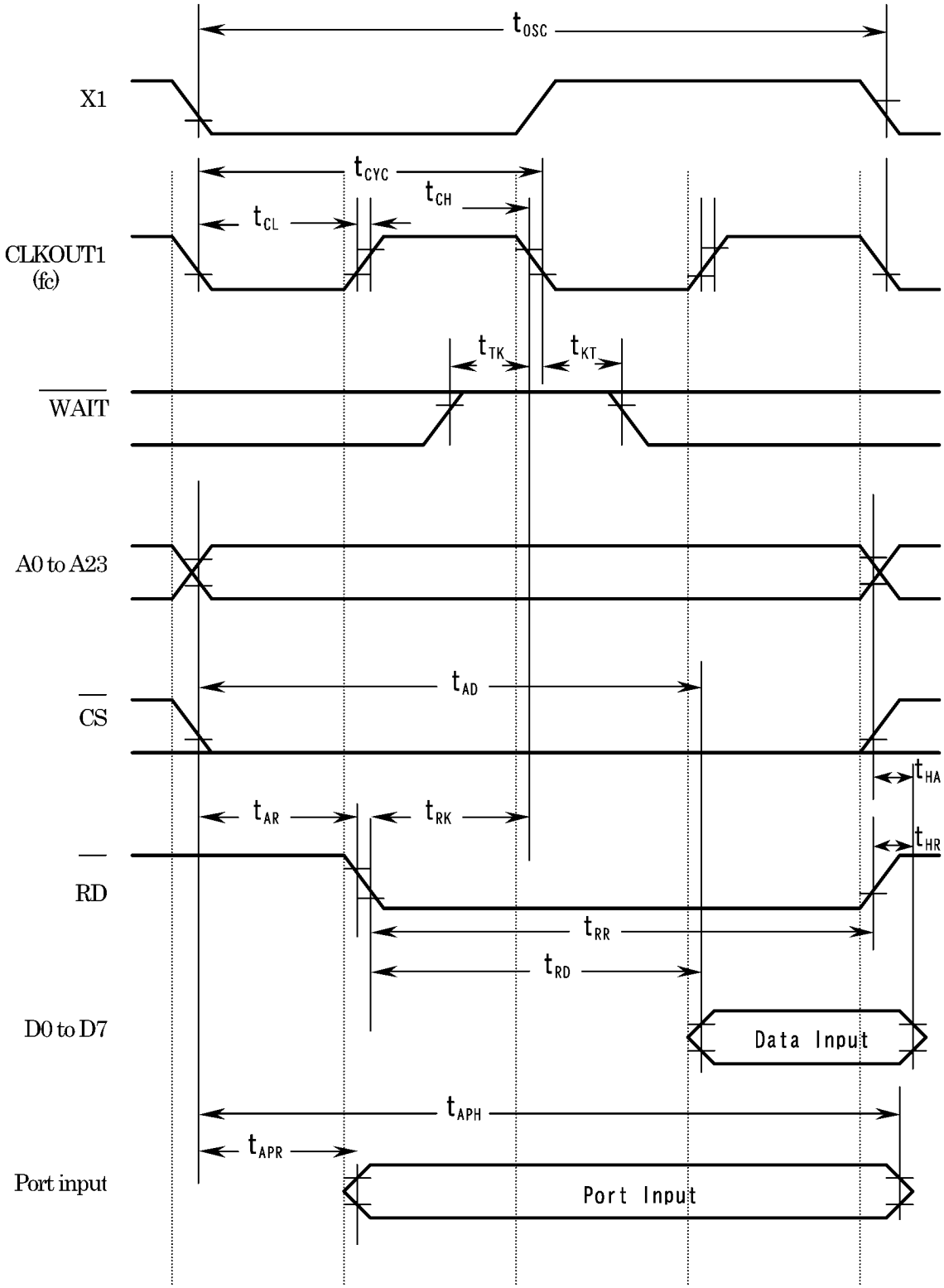
●Input : P0(D0 to D7)

High 2.4V, Low 0.45V, CL=50pF

Others

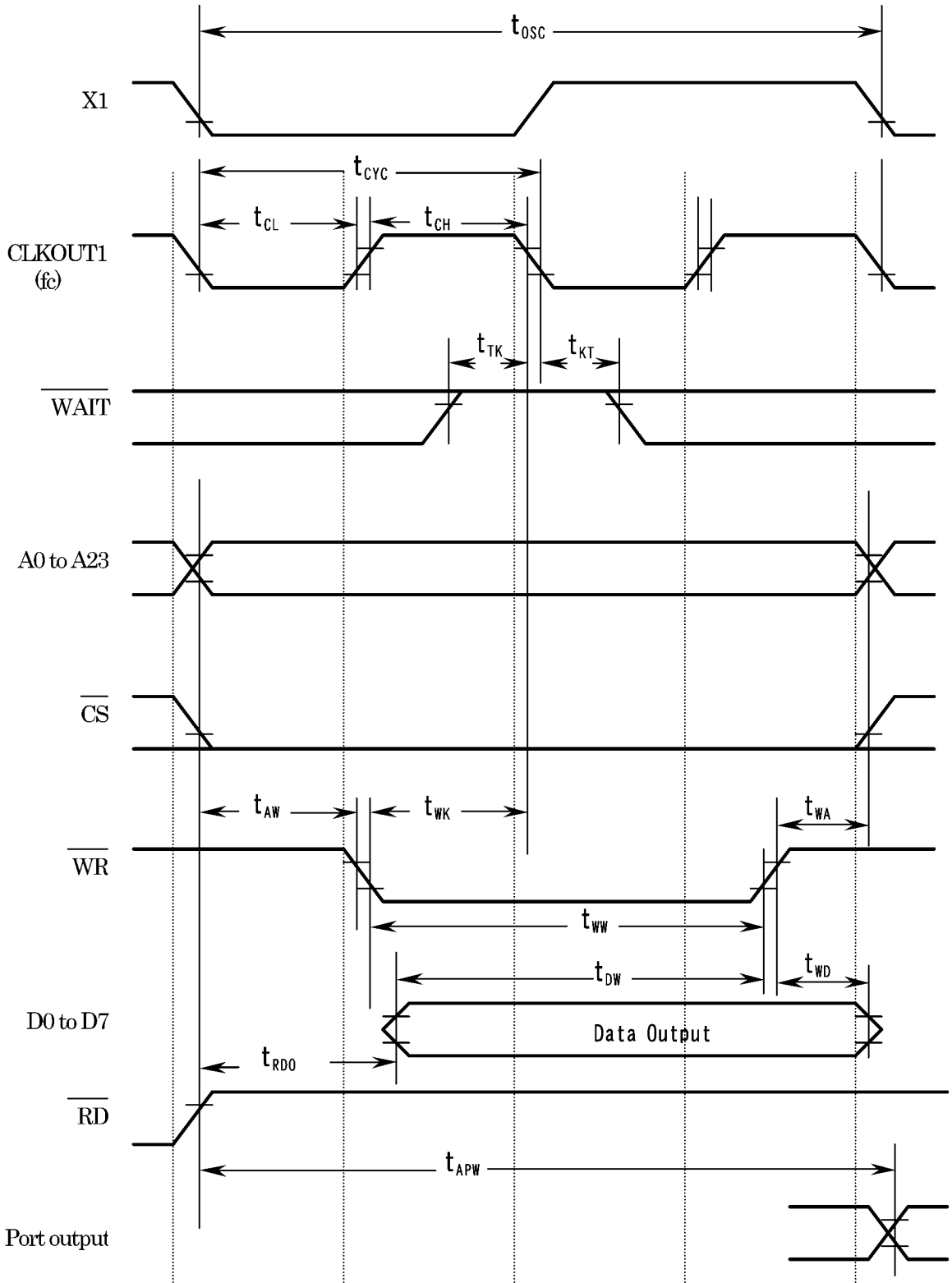
High 0.8×VCC5, Low 0.2×VCC5

(1) Read cycle (0 wait)



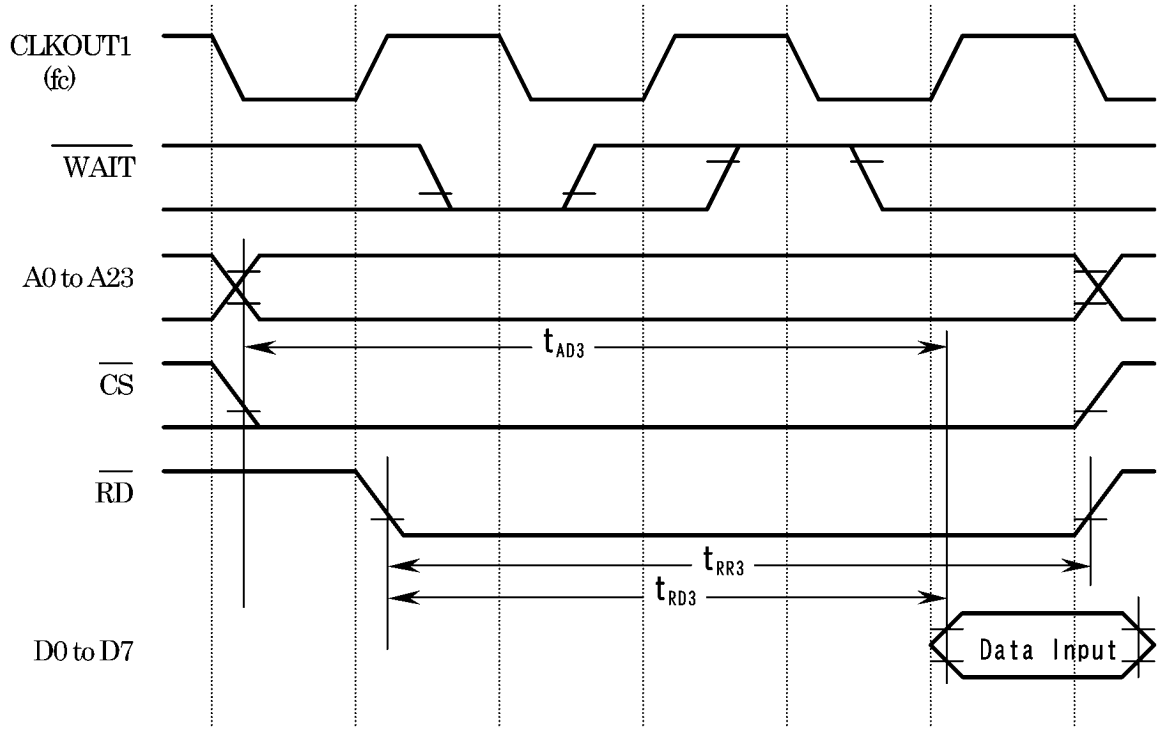
Note : The phase relation between X1 input signal and the other signals is unsettled.
The timing chart above is an example.

(2) Write cycle (0 wait)

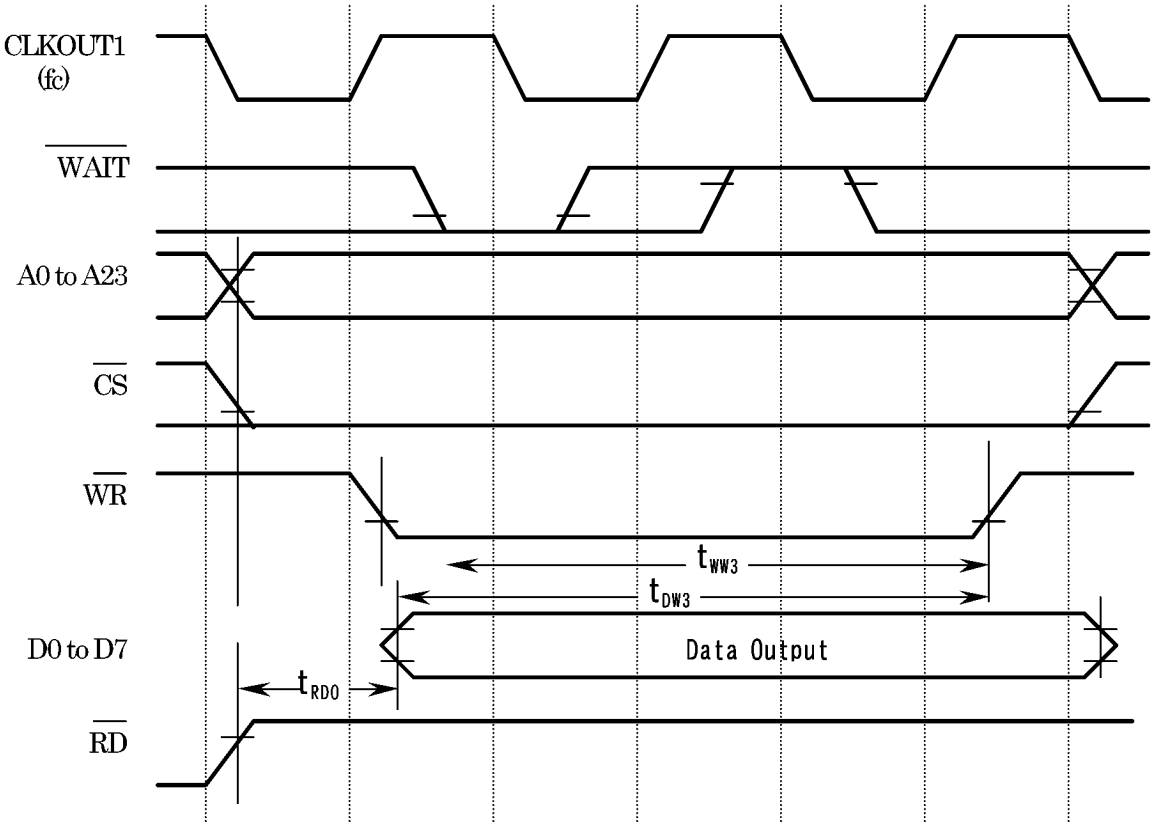


Note : The phase relation between X1 input signal and the other signals is unsettled.
The timing chart above is an example.

(3) Read cycle (1 wait)



(4) Write cycle (1 wait)



4.4 AD Conversion Characteristics

Symbol	Parameter	Min	Typ	MAX	Unit
VREFH	Analog reference voltage (+)	VCC5-0.2	VCC5	VCC5	V
VREFL	Analog reference voltage (−)	VSS5	VSS5	VSS5	
AVCC	AD Converter Power Supply Voltage	VCC5-0.2	VCC5	VCC5	
AVSS	AD Converter Ground	VSS5	VSS5	VSS5	
AVIN	Analog Input Voltage	VREFL		VREFH	
IREF	Analog Current for analog reference voltage		0.8	1.2	mA
E_T	Total error (Quantize error of ±0.5LSB is included)			±3.0	LSB

4.5 Event Counter (TI0, TI4, TI8, TI9, TIA, TIB)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock Cycle	T_{VCK}	8T+100		500		600		ns
Clock Low Width	T_{VCKL}	4T+40		240		290		ns
Clock High Width	T_{VCKH}	4T+40		240		290		ns

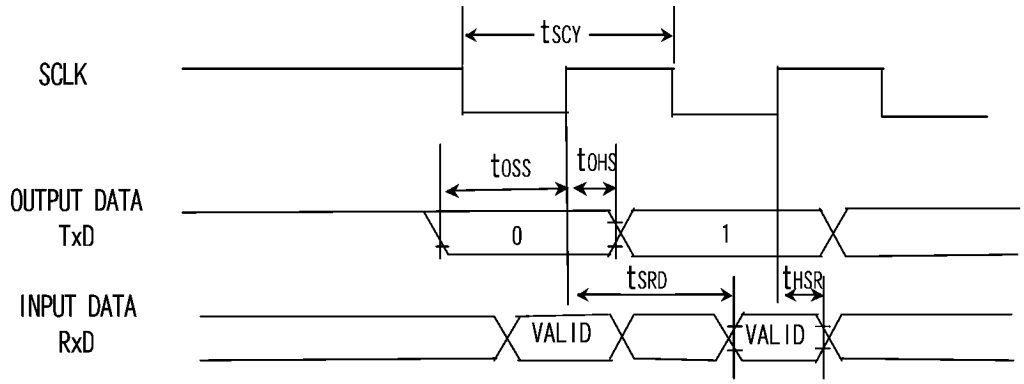
4.6 Serial Channel Timing

(1) SCLK Input mode (I/O Interface mode)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Cycle	T_{SCY}	16T		0.8		1.0		μs
Output Data → SCLK Rise	T_{OSS}	T_{SCY}/2-50		350		450		ns
SCLK Rise → Output Data Hold	T_{OHS}	T_{SCY}/2-100		300		400		
SCLK Rise → Input Data Hold	T_{HSR}	0		0		0		
SCLK Rise → Input Data Valid	T_{SRD}		T_{SCY}/2-100		300		400	

(2) SCLK Output mode (I/O Interface mode)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Cycle (programmable)	T_{SCY}	16T	8192T	0.8	409.6	1.0	512	μs
Output Data → SCLK Rise	T_{OSS}	T_{SCY}/2-150		250		350		ns
SCLK Rise → Output Data Hold	T_{OHS}	T_{SCY}/2-80		320		420		
SCLK Rise → Input Data Hold	T_{HSR}	0		0		0		
SCLK Rise → Input Data Valid	T_{SRD}		T_{SCY}/2-150		250		350	



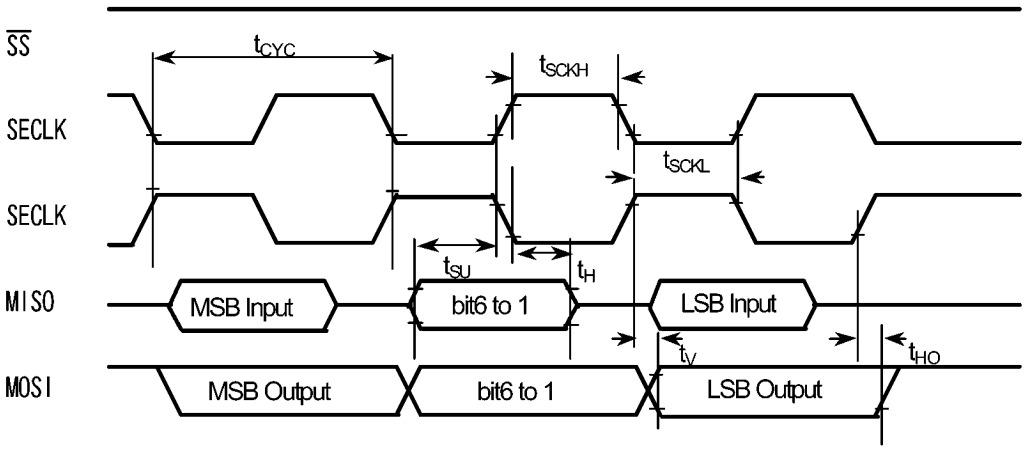
4.7 Interrupt Operation

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 Low Width	T_{INTAL}	4T		200		250		ns
$\overline{NMI}$, INT0 High Width	T_{INTAH}	4T		200		250		
INT1～INT7 Low Width	T_{INTBL}	8T+100		500		600		
INT1～INT7 High Width	T_{INTBH}	8T+100		500		600		

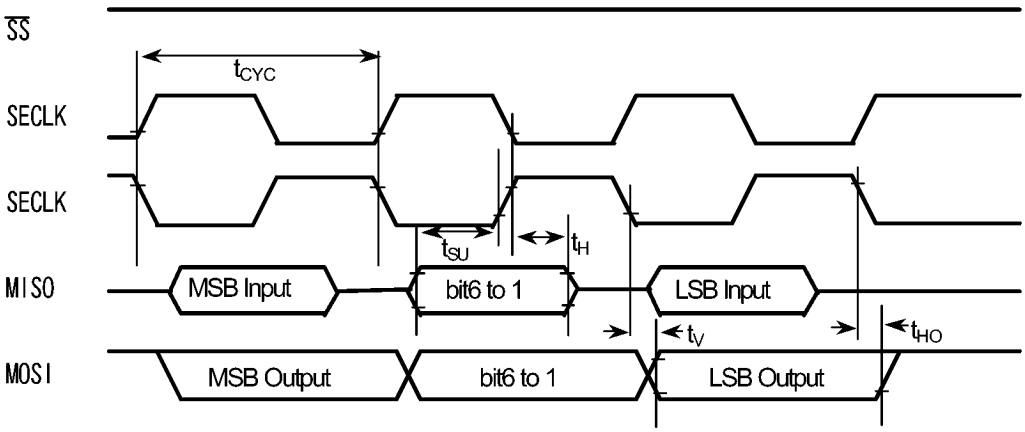
4.8 Serial Expansion Interface

Symbol	Parameter	Variable		20MHz		Unit
		Min	Max	Min	Max	
Tcyc	SECLK Cycle	2.5T	40T	125	2000	ns
Tlead	SS fall → SECLK	2T		100		ns
Tlag	SECLK → SS rise	2T		100		ns
TsckH	SECLK High Pulse Width	Tcyc/2-15		58		ns
TsckL	SECLK Low Pulse Width	Tcyc/2-15		58		ns
Tsu	Input Data Set-up	Tcyc/4		26		ns
Th	Input Data Hold	Tcyc/4		31		ns
Tv	Output Data Valid		Tcyc/4		31	ns
Tho	Output Data Hold	0		0		ns

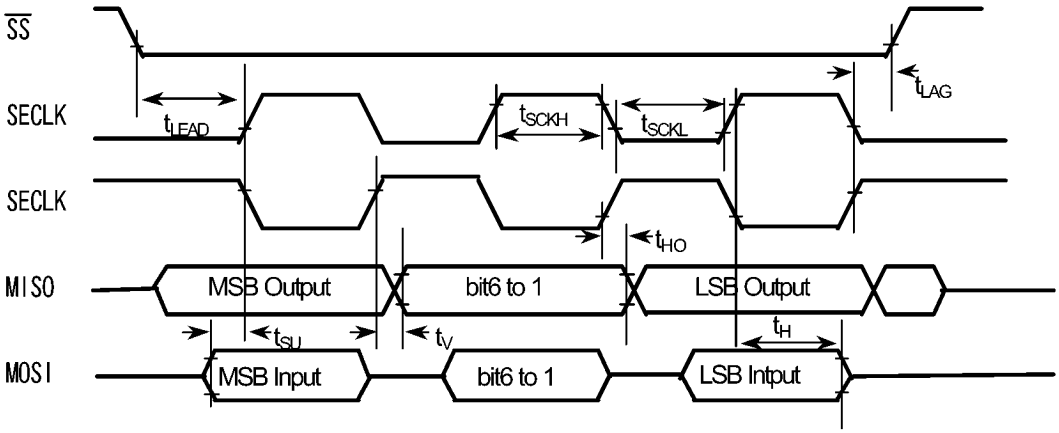
a) SEI Master (CPHA=0)



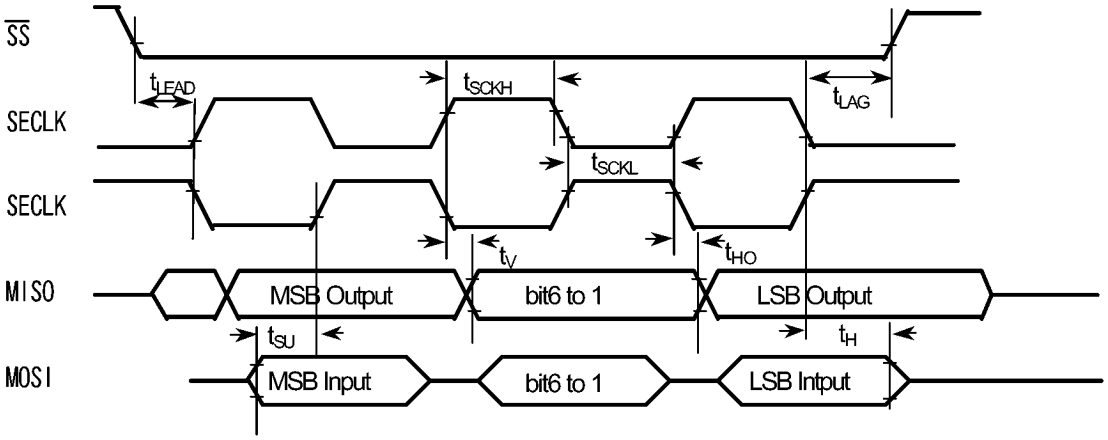
b) SEI Master (CPHA=1)



c) SEI Slave (CPHA=0)



d) SEI Slave (CPHA=1)



5. Package

Package Dimensions : P-LQFP100-1414-0.50C

Unit : mm