CMOS 16-bit Microcontroller

TMP95FY64F

1. TMP95FY64F Basic Specification

1.1 Outline and Feature

TMP95FY64 is high-speed advanced 16-bit microcontroller developed for controlling medium to large-scale equipment. TMP95FY64 has 256K-Byte Flash memory which can be rewritten and erased on board. TMP95FY64 is housed in QFP-100pin package.

Device characteristics are as follows:

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

- TLCS-90/900 instruction mnemonic upward compatible.
- 16M-byte linear address space
- General-purpose registers and register bank system
- 16-bit multiplication/ division and bit transfer/arithmetic instructions
- Micro DMA :4 channels(640ns/2bytes at 25MHz)

(2) Minimum instruction execution time:160ns at 25MHz

(3) Internal RAM:8Kbyte
Internal ROM:256Kbyte Flash memory

(4) External memory expansion

- Can be expanded up to 16M byte (for both programs and data)
- AM8/$\overline{16}$pin (select the external data bus width)
- Can mix 8- and 16-bit external data buses. ······· Dynamic data bus sizing

(5) 8-bit timer:8 channels

- Including event counter function(2 channels)

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

(7) Serial interface:3 channels

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

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

(10) Watchdog timer

(11) Chip select/wait controller:4 blocks

(12) Interrupt functions:45-Interrupt sources

- 9-CPU interrupts ····· SWI instruction, and Illegal instruction
- 26-Internal interrupts·····7-level priority can be set.
- 10-External interrupts·····7-level priority can be set.

(13) I/O ports : Single chip mode 81 pins
Multi chip mode 55 pins(at AM8/$\overline{16}$="H")

(14) Standby function:4 HALT mode(RUN,IDLE2,IDLE1,STOP)

(15) Operating Voltage : Vcc = 4.5 to 5.5V

(16) Package:100pin QFP(LFFP100-P-1414-0.50C:Thickness 2.4mm)

970514EBK1

- TOSHIBA is continually working to improve the quality and the 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 observe standards of safety, and to avoid situations in which a malfuction or failure of a TOSHIBA product 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 products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.
- USP 4,382,279 owned by BULL CP8
- The products described in this document are subject to foreign exchange and foreign trade laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

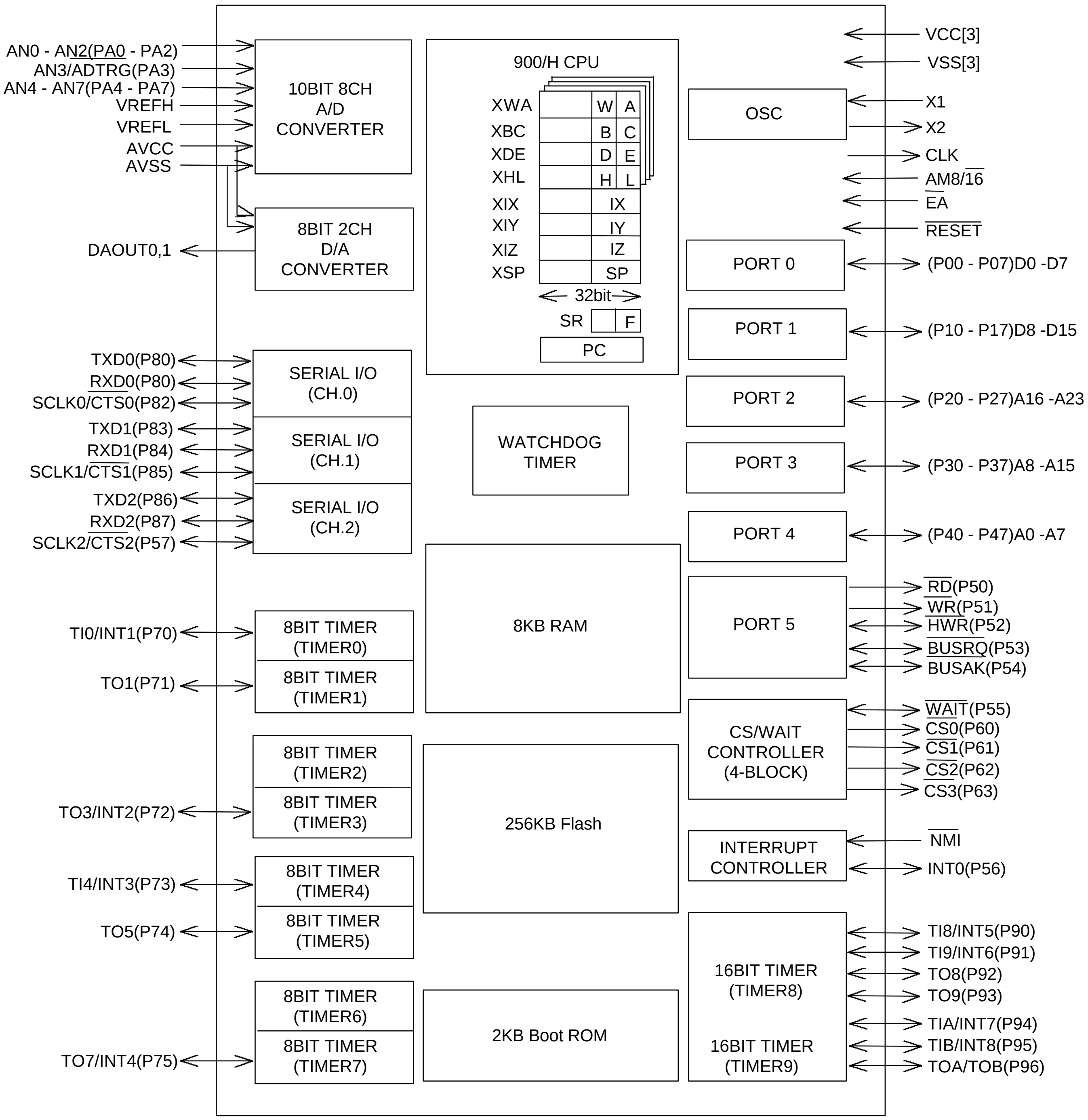
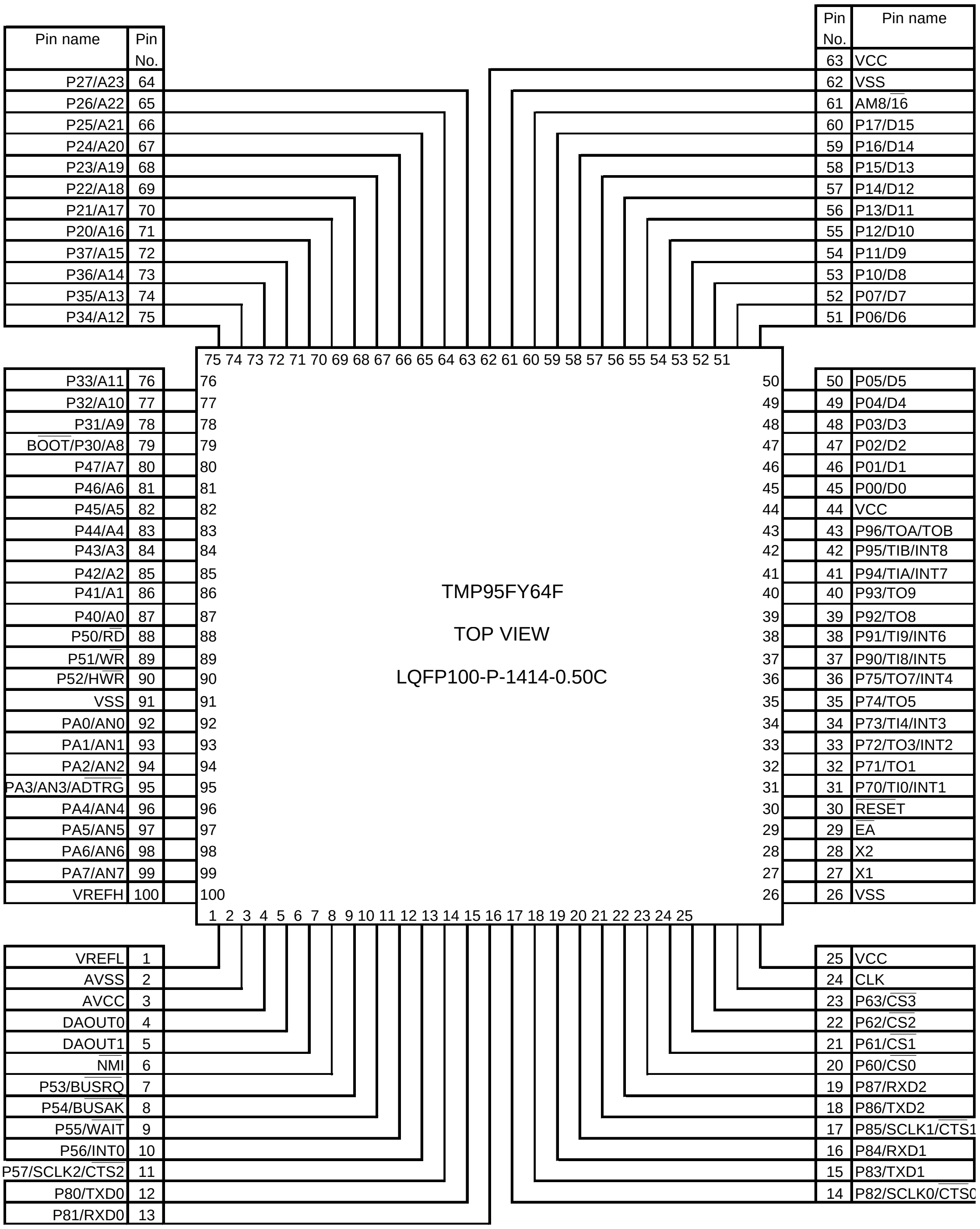


Figure 1 TMP95FY64 Block Diagram

1.2 Pin Assignment and pin functions

1.2.1 Pin Assignment



Pin name	Pin No.
P27/A23	64
P26/A22	65
P25/A21	66
P24/A20	67
P23/A19	68
P22/A18	69
P21/A17	70
P20/A16	71
P37/A15	72
P36/A14	73
P35/A13	74
P34/A12	75

Pin No.	Pin name
63	VCC
62	VSS
61	AM8/$\overline{\text{16}}$
60	P17/D15
59	P16/D14
58	P15/D13
57	P14/D12
56	P13/D11
55	P12/D10
54	P11/D9
53	P10/D8
52	P07/D7
51	P06/D6

P33/A11	76
P32/A10	77
P31/A9	78
$\overline{\text{BOOT}}$/P30/A8	79
P47/A7	80
P46/A6	81
P45/A5	82
P44/A4	83
P43/A3	84
P42/A2	85
P41/A1	86
P40/A0	87
P50/$\overline{\text{RD}}$	88
P51/$\overline{\text{WR}}$	89
P52/$\overline{\text{HWR}}$	90
VSS	91
PA0/AN0	92
PA1/AN1	93
PA2/AN2	94
PA3/AN3/ADTRG	95
PA4/AN4	96
PA5/AN5	97
PA6/AN6	98
PA7/AN7	99
VREFH	100

50	P05/D5
49	P04/D4
48	P03/D3
47	P02/D2
46	P01/D1
45	P00/D0
44	VCC
43	P96/TOA/TOB
42	P95/TIB/INT8
41	P94/TIA/INT7
40	P93/TO9
39	P92/TO8
38	P91/TI9/INT6
37	P90/TI8/INT5
36	P75/TO7/INT4
35	P74/TO5
34	P73/TI4/INT3
33	P72/TO3/INT2
32	P71/TO1
31	P70/TI0/INT1
30	$\overline{\text{RESET}}$
29	$\overline{\text{EA}}$
28	X2
27	X1
26	VSS

VREFL	1
AVSS	2
AVCC	3
DAOUT0	4
DAOUT1	5
$\overline{\text{NMI}}$	6
P53/$\overline{\text{BUSRQ}}$	7
P54/$\overline{\text{BUSAK}}$	8
P55/$\overline{\text{WAIT}}$	9
P56/INT0	10
P57/SCLK2/$\overline{\text{CTS2}}$	11
P80/TXD0	12
P81/RXD0	13

25	VCC
24	CLK
23	P63/$\overline{\text{CS3}}$
22	P62/$\overline{\text{CS2}}$
21	P61/$\overline{\text{CS1}}$
20	P60/$\overline{\text{CS0}}$
19	P87/RXD2
18	P86/TXD2
17	P85/SCLK1/$\overline{\text{CTS1}}$
16	P84/RXD1
15	P83/TXD1
14	P82/SCLK0/$\overline{\text{CTS0}}$

TMP95FY64 Pin Assignment

2.2 Pin name and functions

The names of input/output pins and their functions are described below.

Table 2.2 Pin names and functions.

Table 2.2 Pin names and functions(1/4)

Pin name	Number of pins	I/O	functions
P00 to P07	8	I/O	Port 0 :I/O ports that allow I/O to be selected on a bit basis.
/D0 to D7		I/O	Data :0 to 7 for data bus
P10 to P17	8	I/O	Port 1 :I/O ports that allow I/O to be selected on a bit basis.
/D8 to D15		I/O	Data :8 to 15 for data bus
P20 to P27	8	I/O	Port 1 :I/O ports that allow I/O to be selected on a bit basis.
/A16 to A23		Output	Address :16 to 23 for address bus
P30	1	I/O	Port 30 : I/O port
/A8		Output	Address : 8 for address bus
/$\overline{\text{BOOT}}$		Input	Setting pin for Single BOOT mode. Pull-up with external resister.
P31 to P37	7	I/O	Port 3 :I/O ports that allow I/O to be selected on a bit basis.
/A9 to A15		Output	Address :9 to 15 for address bus
P40 to P47	8	I/O	Port 4 :I/O ports that allow I/O to be selected on a bit basis.
/A0 to A7		Output	Address :0 to 7 for address bus
P50	1	Output	Port 50 :Output port
/$\overline{\text{RD}}$		Output	Read :Strobe signal for reading external memory. (When P5<P50>=0, P5FC<P50F>=1, strobe signal output at all read timing.)
P51	1	Output	Port 51 :Output port
/$\overline{\text{WR}}$		Output	Write :Strobe signal for writing data on pins D0 to 7.
P52	1	I/O	Port 52 :I/O port(with pull-up resistor)
/$\overline{\text{HWR}}$		Output	High Write :Strobe signal for writing data on pins D8 to 15.
P53	1	I/O	Port 53 :I/O port(with pull-up resistor)
/$\overline{\text{BUSRQ}}$		Input	Bus request :Signal used to request bus release to external bus.
P54	1	I/O	Port 54 :I/O port(with pull-up resistor)
/$\overline{\text{BUSAK}}$		Output	Bus acknowledge :Signal indicating bus release.
P55	1	I/O	port 55 : I/O port(with pull-up resistor)
/$\overline{\text{WAIT}}$		Input	Wait :Pin used to request CPU bus wait.
P56	1	I/O	Port 56 : I/O port(with pull-up resistor)
/INT0		Input	Interrupt request pin 0:Interrupt request pin with programmable level/rising edge.

Table 2.2 Pin names and functions(2/4)

Pin name	Number of pins	I/O	functions
P57	1	I/O	Port 57:I/O port(with pull-up resistor)
/SCLK2		I/O	Serial Clock I/O 2
/$\overline{\text{CTS2}}$		Input	Serial data send enable 2(Clear To Send)
P60	1	Output	Port 60 :Output port
/$\overline{\text{CS0}}$		Output	Chip select 0:Output 0 when address is within specified address.
P61	1	Output	Port 61 :Output port
/$\overline{\text{CS1}}$		Output	Chip select 1:Output 0 when address is within specified address.
P62	1	Output	Port 62 :Output port
/$\overline{\text{CS2}}$		Output	Chip select 2:Output 0 when address is within specified address.
P63	1	Output	Port 63 :Output port
/$\overline{\text{CS3}}$		Output	Chip select 3:Output 0 when address is within specified address.
P70	1	I/O	Port 70:I/O port
/TI0		Input	Timer input 0
/INT1		Input	Interrupt request pin 1:Interrupt request pin with rising edge.
P71	1	I/O	Port 71:I/O port
/TO1		Output	Timer out 1:Timer 0 or Timer 1 output
P72	1	I/O	Port 72:I/O port
/TO3		Output	Timer output 3:Timer 2 or Timer 3 output
/INT2		Input	Interrupt request pin 2:Interrupt request pin with rising edge.
P73	1	I/O	Port 73:I/O port
/TI4		Input	Timer input 4:Timer 4 input
/INT3		Input	Interrupt request pin 3:Interrupt request pin with rising edge.
P74	1	I/O	Port 74:I/O port
/TO5		Output	Timer output 5:Timer 4 or Timer 5 output
P75	1	I/O	Port 75:I/O port
/TO7		Output	Timer output 7:Timer 6 or Timer 7 output
/INT4		Input	Interrupt request pin 4:Interrupt request pin with rising edge.
P80	1	I/O	Port 80:I/O port(with pull-up resistor)
/TXD0		Output	Serial send data 0
P81	1	I/O	Port 81:I/O port(with pull-up resistor)
/RXD0		Input	Serial receive data 0
P82	1	I/O	Port 82:I/O port(with pull-up resistor)
/SCLK0		I/O	Serial Clock I/O 0
/$\overline{\text{CTS0}}$		Input	Serial data send enable 0(Clear To Send)

Table2.2 Pin names and function(3/4)

Pin name	Number of pins	I/O	function
P83	1	I/O	Port 83:I/O port(with pull-up resistor)
/TXD1		Output	Serial send data 1
P84	1	I/O	Port 84:I/O port(with pull-up resistor)
/RXD1		Input	Serial receive data 1
P85	1	I/O	Port 85:I/O port(with pull-up resistor)
/SCLK1		I/O	Serial Clock I/O 1
/$\overline{\text{CTS1}}$		Input	Serial data send enable 1(Clear To Send)
P86	1	I/O	Port 86:I/O port(with pull-up resistor)
/TXD2		Output	Serial send data 2
P87	1	I/O	Port 87:I/O port(with pull-up resistor)
/RXD2		Input	Serial receive data 2
P90	1	I/O	Port 90:I/O port
/TI8		Input	Timer input 8:Timer 8 input
/INT5		Input	Interrupt request pin 5:Interrupt request pin with programmable rising/falling edge.
P91	1	I/O	Port 91:I/O port
/TI9		Input	Timer input 9:Timer 8 input
/INT6		Input	Interrupt request pin 6:Interrupt request pin with rising edge.
P92	1	I/O	Port 92:I/O port
/TO8		Output	Timer output 8:Timer 8 output
P93	1	I/O	Port 93:I/O port
/TO9		Output	Timer output 9:Timer 8 output
P94	1	I/O	Port 94:I/O port
/TIA		Input	Timer input A :Timer 9 input
/INT7		Input	Interrupt request pin 7:Interrupt request pin with programmable rising/falling edge.
P95	1	I/O	Port 95:I/O port
/TIB		Input	Timer input B :Timer 9 input
/INT8		Input	Interrupt request pin 8:Interrupt request pin with rising edge.
P96	1	I/O	Port 96:I/O port
/TOA		Output	Timer output A :Timer 9 output
/TOB		Output	Timer output B :Timer 9 output
PA0 to PA2	3	Input	Port A0 to A2:Input port
/AN0 to AN2		Input	Analog input 0 to 2:Input to A/D converter
PA3	1	Input	Port A3:Input port
/AN3		Input	Analog input 3:Input to A/D converter
/$\overline{\text{ADTRG}}$		Input	External A/D conversion start trigger input

Table 2.2 Pin names and function(4/4)

Pin name	Number of pin	I/O	function
PA4 to PA7	4	Input	Port A4 to A7:Input port
/AN4 to AN7		Input	Analog input 4 to 7:Input to A/D converter
DAOUT0	1	Output	D/A output 0:D/A converter 0 analog current output pin
DAOUT1	1	Output	D/A output 1:D/A converter 1 analog current output pin
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin :Interrupt request pin with programmable falling/both edge.
CLK	1	Output	Clock output :Outputs external input clock X1 divided by 4. Pulled up during reset.
$\overline{\text{EA}}$	1	Input	External access :Connect to Vcc when single chip mode. Connect to GND when multi chip mode.
AM8/$\overline{\text{16}}$	1	Input	Address mode:External data bus width selection pin. Set to 0 when using fixed 16-bit external bus or dual 8/16-bit external bus. Set to 1 with 8-bit external bus fixed.
RESET	1	Input	Reset:Initializes LSI(with pull-up resistor)
VREFH	1	Input	Reference voltage input pin for A/D converter (H)
VREFL	1	Input	Reference voltage input pin for A/D converter (L)
AVCC	1		A/D and D/A converter power supply pin
AVSS	1		A/D and D/A converter ground pin (0V)
X1 / X2	2	Input/Output	Oscillator connecting pins
VCC3			Power supply pin (+5V)
VSS	3		Ground pin (0V)

Note 1: Apart from $\overline{\text{RESET}}$ pin, the pull-up resistors can be disconnected by software.

Note 2: Connect all VCC and AVCC pins to power supply and all VSS and AVSS pins to GND.

3. Operation

The TMP95FY64 is the MCU which includes 256K byte Flash ROM and 8K byte RAM, and has operates as the same way as the TMP95CS64 which includes 64K byte Mask ROM and 2K byte RAM.

Please refer to the TMP95CS64 data sheets for the function not described here.

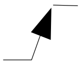
3.1 Operation mode

The TMP95FY64 has a single chip mode and a single boot mode. Each mode is set by pin status after reset.

- Single Chip Mode: Normal operating mode. The device starts executing program on internal Flash memory after reset.

- Single Boot Mode: Internal Flash memory re-programming mode with serial(UART) interface. Internal boot ROM starts and on-board re-write program is executed after reset.

Table 3.1(1) Operation mode setting table

Operation Mode	mode setting input pin			
	$\overline{\text{RESET}}$	CLK	$\overline{\text{BOOT}}$	$\overline{\text{EA}}$
Single Chip		Open(Pulled-up during reset)	H	H
Single Boot			L	

3.2 Memory Map

The memory map and capacity of built in Flash ROM and RAM are different from TMP95CS64.

The figure 3.2(1) shows memory map and CPU addressing mode area on single chip mode.

The figure 3.2(2) shows memory map on each operating mode.

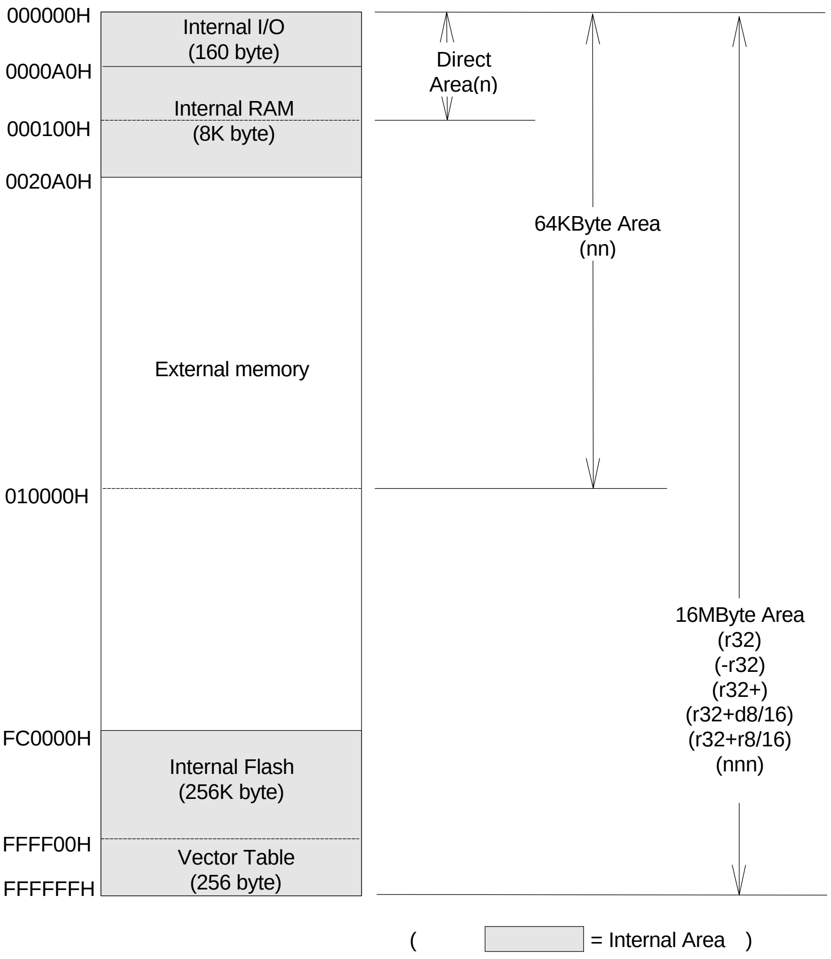


Figure 3.2(1)TMP95FY64 memory map(Single Chip Mode)

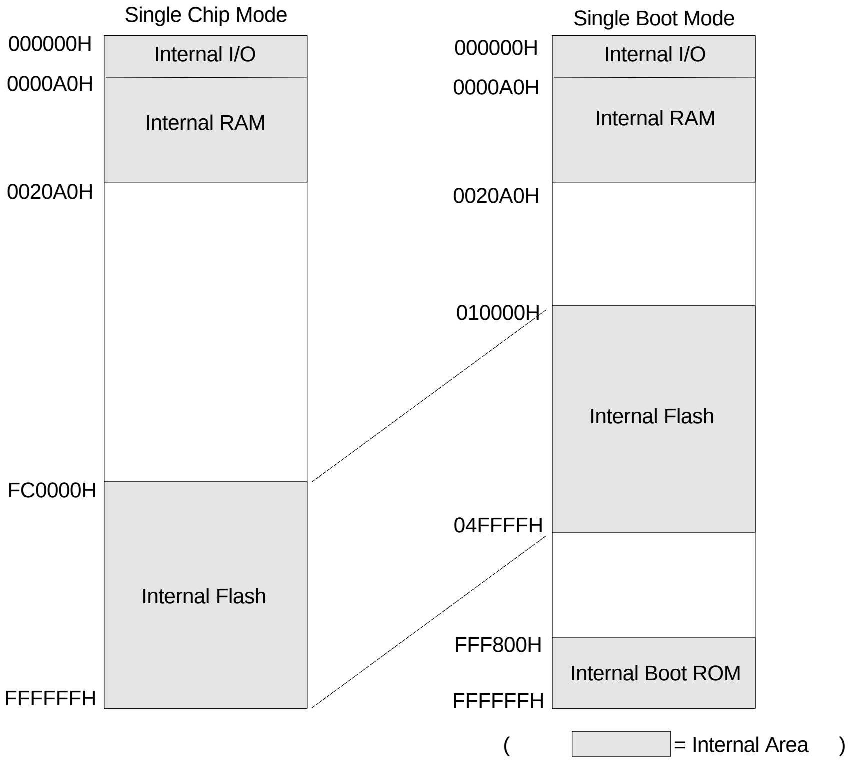


Figure 3.2(2)TMP95FY64 memory map on each operating mode

3.3 Flash memory

The TMP95FY64 has Flash memory which can be erased and Programmed on 5V single voltage.

Erase and program of Flash memory are operated by JEDEC standard command. After command input, programming and erasing are automatically operated. The erase function has a chip erase, a block erase, and a plural block erase.

Feature:

- Program/Erase voltage
  Vcc=5V +/-10%
- Configuration
  256K x 8 bit/128K x 16 bit(256K byte)
- Function
  Auto-Programming
  Auto-Chip erasing
  Auto-Block erasing
  Auto-Multi-Block erasing
  Data Polling/Toggle bit
- Block erasing architecture
  16K byte x 1/8K byte x 2/
  32K byte x 1/64K byte x 3
- Mode control
  JEDEC Standard command
- Flash memory Type
  29F200T

Block protect/ID read are not supported.

Block Configuration:

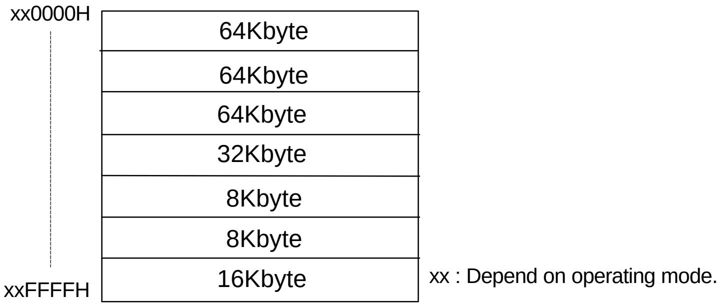


Figure 3.3(1)Block configuration of Flash memory

Internal interface:

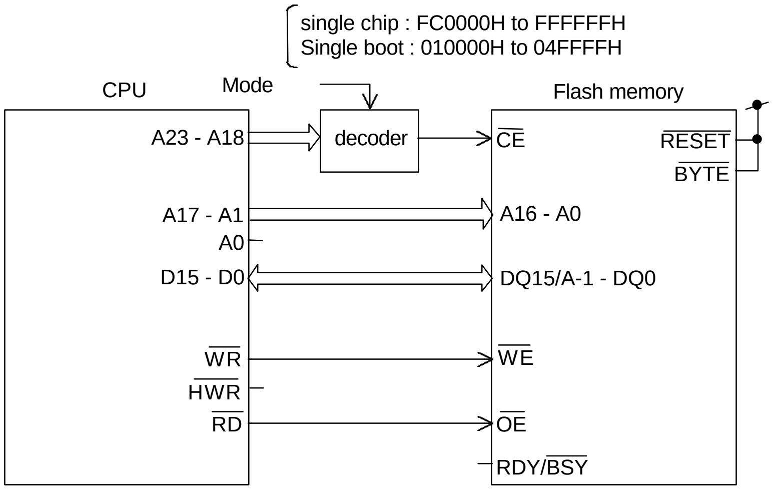


Figure 3.3(2)Flash memory internal interface

Command Sequence :Flash memory access by internal CPU

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

CPU address

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
x X X X X H	Flash memory address area	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
x A A A A H		1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
x 5 5 5 4 H		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 read by each byte or word.

PA:Program address
PD:Program data input
data write by each even address/word.

BA:Block address

Hardware sequence flag :flash memory access by internal CPU

status		DQ7	DQ6	DQ5	DQ3
executing automatic operation.	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, DQ0to DQ2 are Don't care.

Block erase address table :Flash memory access by internal CPU

Block	address					Address area		Size
	A17	A16	A15	A14	A13	Single chip	Single boot	
BA0	L	L	x	x	x	FC0000H - FCFFFFH	010000H - 01FFFFH	64K byte
BA1	L	H	x	x	x	FD0000H - FDFFFFH	020000H - 02FFFFH	64K byte
BA2	H	L	x	x	x	FE0000H - FEFFFFH	030000H - 03FFFFH	64K byte
BA3	H	H	L	x	x	FF0000H - FF7FFFH	040000H - 047FFFH	32K byte
BA4	H	H	H	L	L	FF8000H - FF9FFFH	048000H - 049FFFH	8K byte
BA5	H	H	H	L	H	FFA000H - FFBFFFH	04A000H - 04BFFFH	8K byte
BA6	H	H	H	H	x	FFC000H - FFFFFFH	04C000H - 04FFFFH	16K byte

Basic operation: Flash memory access with internal CPU

This flash memory has two kinds of operation modes of the reading mode and the automatic operation mode roughly dividing. It is possible to move to the automatic operation mode by executing the command sequence in the reading mode. The automatic operation mode inside cannot read the memory data.

(1) Reading

When data is read, the flash memory is set in the reading mode. The flash memory becomes a reading mode at immediately after the power supply turning on and the normal termination of an automatic operation.
The reset command described later is used for the return to the reading mode from abnormal termination of an automatic operation, and other modes.

(2) Command writing

This flash memory uses the method of the JEDEC standard command control. Writing in the command register is operated by executing the command sequence for the flash memory. The flash memory latches input address and data to the command register, and executes the operation corresponding to the command.
The input of the command data uses DQ0-DQ7. The input of DQ8-DQ15 is disregarded. To cancel the command input while inputting the command sequence, the reset command is input. If the reset command is accepted, the flash memory resets the command register, and becomes a reading mode. Moreover, when a wrong command sequence is input, the flash memory resets the command register, and becomes a reading mode.

(3) Reset (reset command)

The flash memory does not return to the reading mode when an automatic operation terminates abnormally.
In this case, the flash memory is returned to the reading mode by reading/the reset command. Moreover, when the command input on the way is canceled, the content of the command register can be cleared by reading/the reset command.

(4) Automatic program

It is necessary to be written in the flash memory every even number address/word byte. The automatic program operation latches the program address/the program data every even number address/word byte at the fourth bus writing cycle of the command cycle. An automatic program begins at time when the program data was latched. When the operation begins, the program and the program verify are automatically operated internally. The operation of an automatic program can be confirmed with the hardware sequence flag.
The automatic program operation inside does not accept the input of the command sequence. "1" data cell can be made "0" data by writing in the flash memory. However, "0" data cell cannot be made "1" data. "0" data cell can be made "1" data by the erase operation. When the automatic program operation becomes defective, the flash memory is locked like this mode, and does not return to the reading mode. This state can be confirmed with the hardware sequence flag. It is necessary to reset the flash memory by the reset command. In this case, writing in this address should be defective, and the block which includes this address thereafter not be used.

(5) Automatic chip erase

The automatic chip erase operation begins from the sixth bus writing cycle end at the command cycle.
When the operation begins, preprogramming all the addresses to "0" data are executed automatically in the flash memory, and continuously, the erase and erase verify are executed. The state of the automatic chip erase operation can be confirmed with the hardware sequence flag. The automatic chip erase operation inside does not accept the command sequence input. When the automatic erase operation becomes defective, the flash memory is locked like this mode, and does not return to the reading mode. This state can be confirmed with the hardware sequence flag. Please reset the flash memory by the reset command. Moreover, the block where the defect occurs cannot be detected. It is necessary to stop the use of the device or not to use a defective block detected by the block erase.

(6) Automatic block erase/automatic multi block erase

The automatic block erase begins later in time of the erase holding from the sixth bus writing cycle end at the command cycle. When the operation begins, preprogramming all the addresses in selected blocks to "0" data are executed automatically in the flash memory, and continuously, the erase and erase verify are executed. When do the erasure of the plural block, each block address and the automatic block erase command are input in the erase hold time repeating the sixth bus writing cycle. When the command sequences other than the automatic block erase are input during the erase hold time, the flash memory is reset, and becomes a reading mode. The erase hold time is 50us. The counting for the erase hold time is begun at each end of the sixth bus writing cycle. The state of the automatic block erase operation can be confirmed with the hardware sequence flag. The input of other command sequences is not accepted during the automatic block erase. When the automatic block erase operation becomes defective, the flash memory is locked like this mode, and does not return to the reading mode. This state can be confirmed with the hardware sequence flag. Please reset the flash memory by the reset command. When the plural block is selected, the block where the defect occurs cannot be detected. The use of the device is discontinued or it is necessary to do the block erasure individually, specify a defective block, and not use a defective block.

(7) Hardware sequence flag

The automatic operation execution of the flash memory can be confirmed with the hardware sequence flag. Data can be read while operating automatically according to the same timing as the reading mode. The flash memory automatically returns to the reading mode when an automatic operation is ended. The state of the operation can be confirmed during the automatic operation execution with the hardware sequence flag. Moreover, the automatic operation end can be confirmed by the read data's having matched with than the cell data.

1)DQ7 (DATA polling)

An automatic operation of the flash memory can be confirmed by the DATA polling function. The output of the DATA polling begins from end of the last bus writing cycle in the automatic operation command sequence. The automatic program operation inside outputs the reversing data of the data written in DQ7, and after an automatic program ends, outputs the cell data of DQ7. It is possible to identify the state of operation by reading DQ7. The automatic erase operation inside outputs "0" from DQ7, and after this operation ends, outputs "1" (cell data). Moreover, when the result of an automatic operation is defective, DQ7 outputs the automatic operation data continuously. When data is read, it is necessary to give a written address or an arbitrary block address under the erasure because the flash memory releases the address latch when the operation ends.

2)DQ6 (toggle bit)

In addition to the DATA polling, The toggle bit output function is provided as a method of recognizing the state of an automatic operation. The output of the toggle begins from end of the last bus writing cycle in the automatic operation command sequence. However, the output of the toggle in the automatic block erase operation begins after end of the erase hold time. This toggle outputs to DQ6, and outputs the data of "1" and "0" alternately every reading cycle. When an automatic operation ends, DQ6 stops the output of the toggle, and outputs the cell data. When the result of an automatic operation is defective, DQ6 continues the toggle output.

3)DQ5 (internal timer excess)

When an automatic operation is normally done, the flash memory outputs "0" to DQ5. The output of DQ5 changes into "1" if the time for which an automatic operation specify in the flash memory is exceeded. This has the possibility that it is shown that an automatic operation did not end normally, and the flash memory is defective. However, when the "1" data is written in the "0" data cell, DQ5 outputs "1", and the flash memory is judged to be defective. ("1" data cell can be made "0" data by writing in the flash memory. However, "0" data cell cannot be made "1" data.) In this case, DQ5 does not show the defect of the flash memory . It is shown that use is not correct. The flash memory is locked, and does not return to the reading mode when an automatic operation does not end normally. Please reset the flash memory by the reset command.

4)DQ3 (block erase timer)

The automatic block erase begins later in time of the erase holding(50us) from the sixth bus writing cycle end at the command cycle. The flash memory outputs "0" to DQ3 during the block erase hold time, and outputs "1" to DQ3 When the erasure begins. When the erase block is added, inputs the additional block erase command during the block erase hold time for the previous block erase. The block erase hold time is reset whenever the block erase command is input, and the flash memory counts a hold time from the beginning. When the result of an automatic operation is defective, DQ3 outputs "1".

5)RDY/BSY (Ready/Busy)

* This function cannot be used because of no connection with internal CPU.

(8) Flash memory rewriting with internal CPU

The flash memory rewriting with internal CPU is done with the above-mentioned command sequence and a hardware sequence flag. However, the memory data is not able to read from the internal flash memory during the automatic operation mode. It is necessary to execute the rewriting program on the outside area of the flash memory. There are two methods of flash memory rewriting with internal CPU. It is a method of using the single boot mode prepared beforehand, and a method of using original user's protocol on the single chip mode (user boot).

1) Single boot:

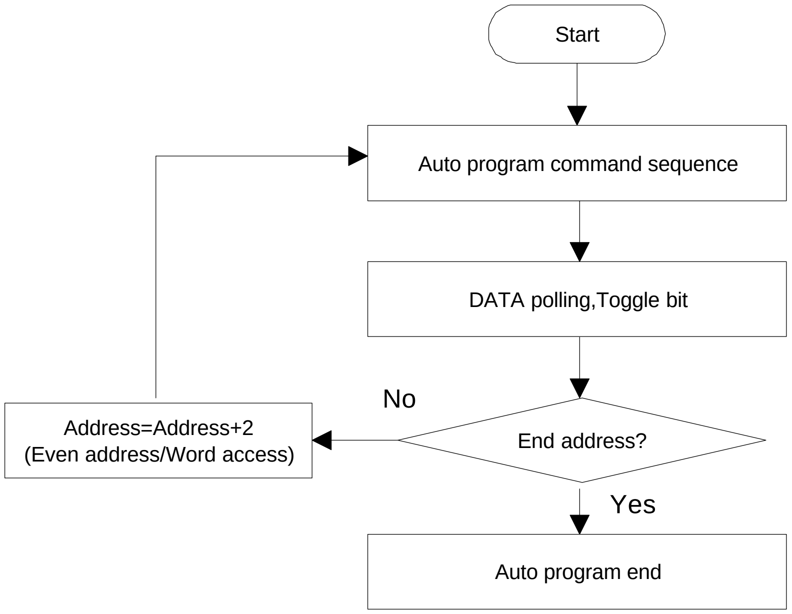
It is a method of rewriting the flash memory by built-in boot ROM program by starting the microcomputer in the single boot mode. In this mode, the built-in boot ROM is mapped in the area where contains the vector table of interrupt, and the boot ROM program is executed. Moreover, the flash memory is mapped in the different area from the boot ROM. The boot ROM program operates receiving the rewrite data by the serial transfer, and rewriting the flash memory. Single boot is done with the interrupt prohibited. The non maskable interrupt(NMI etc.) also must to be prohibited. Please refer to 3.4 single boot mode for details.

2) User boot:

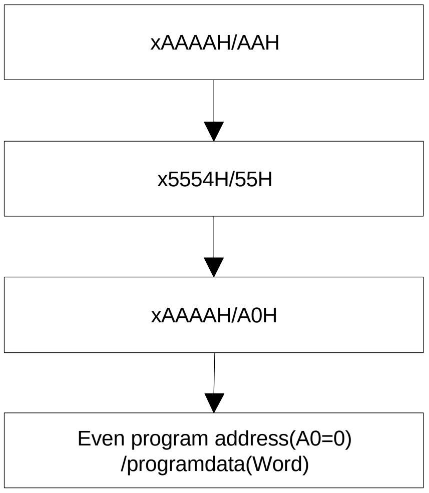
It is a method of using the original user's flash memory rewriting program. This mode is operated in the single chip mode (usual operation mode). This mode should execute the flash memory rewriting program on the different address space from the flash memory area. Moreover, it is necessary to prohibit all the interrupt including the non maskable interrupt as same as single boot. The flash memory rewriting program is prepared beforehand including the data taking routine for rewriting and the flash memory rewriting routine .

In a main program, changing from the usual operation to the flash memory rewriting operation. The flash memory rewriting program which was prepared is transferred to outside the flash memory area and executing. For example, the flash memory rewriting program is transferred from on the flash memory to built-in RAM and executing. Preparing it in an external memory and executing.

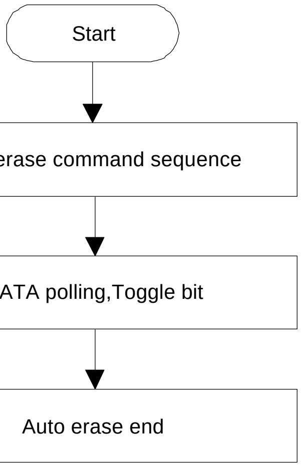
Auto program



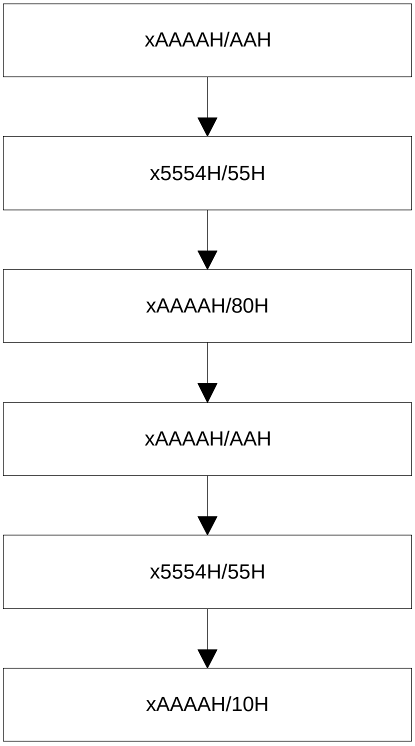
Auto program command sequence(address/command)



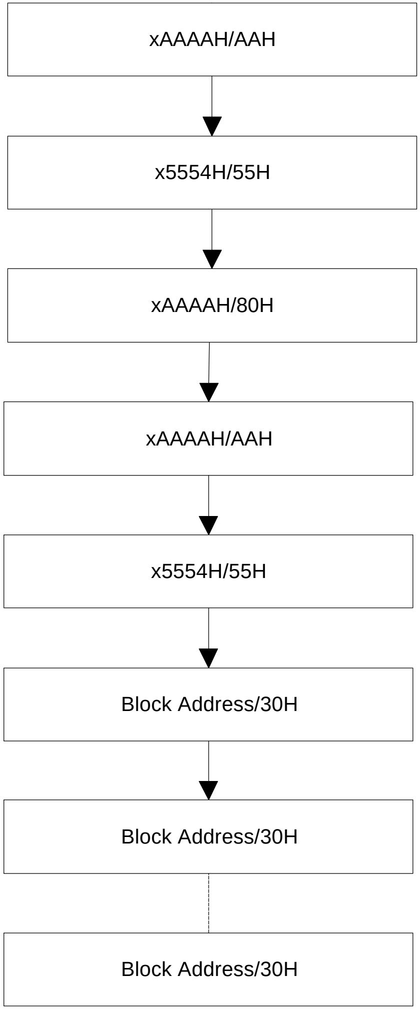
Auto erase



Auto Chip erase command sequence
(Address/Command)

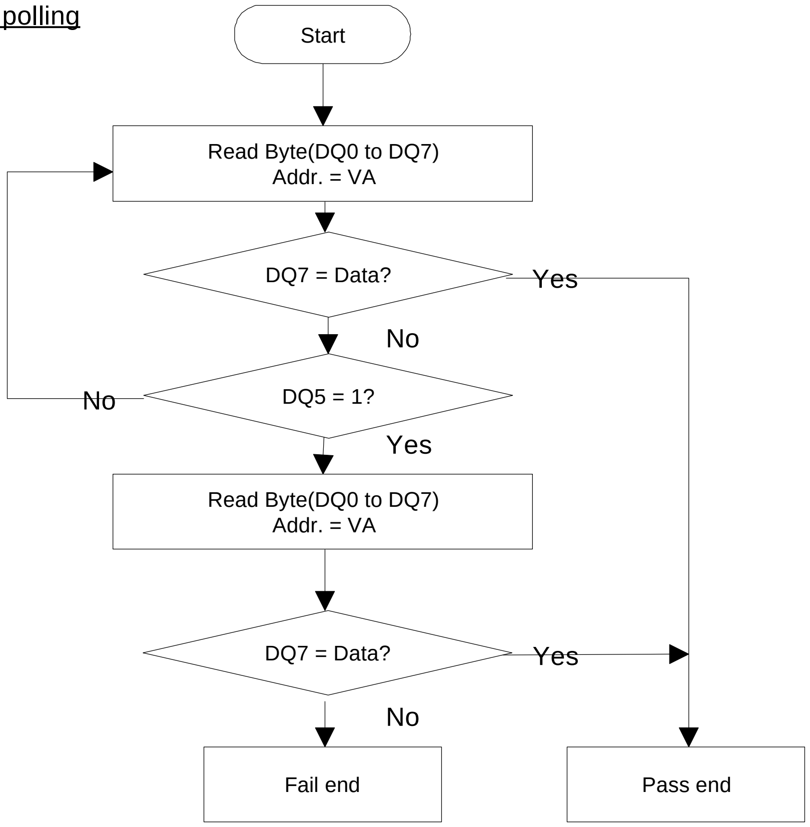


Auto Block /Multi Block erase command sequence
(Address/Command)

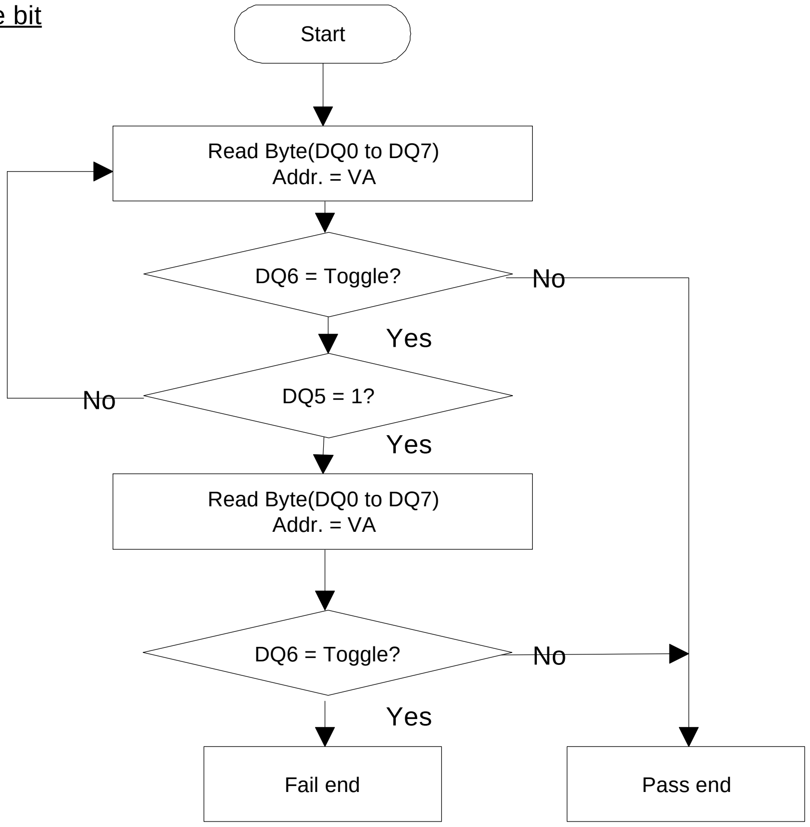


Address input at
auto multi block erase
(under 50us each)

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) Overview

TMP95FY64 has the single boot mode as an operation mode to do the on board programming. The Boot ROM is mapped on the memory space when setting in the single boot mode. The Boot ROM is a mask ROM which does the flash memory rewriting on board. The on board programming is executed by connecting SIO of TMP95FY64 (channel 2) and writing TOOL (controller), and sending the command from the controller side.

Moreover, the loader function to transfer the program data from the outside to built-in RAM of the TMP95FY64 is provided in the boot program built into boot ROM.

Figure 3.4(1) shows the example of connecting the writing controller and the target board.

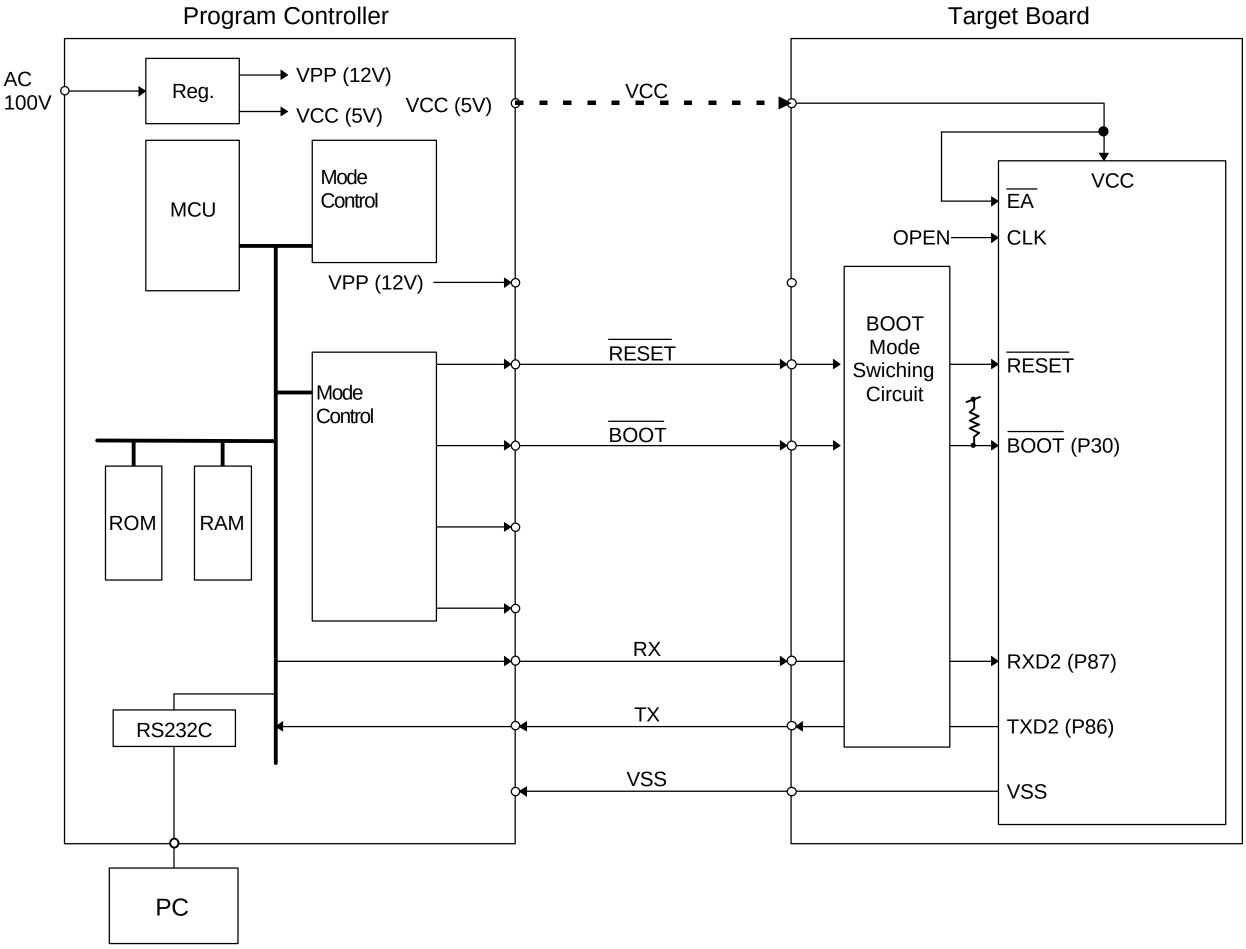


Figure 3.4(1) Example of On-Board Programming Connection

Note 1: The AF200(advanced on-board flash microcomputer programmer) made by Yokogawa Digital Computer (YDC) supports on-board programming for this device. Refer to the AF200 manual for details.

Contact: Yokogawa Digital Computer Corporation
Micom-System Business Group
Instrument Business Division
tel : 81-423-33-6224

(2) Mode setting

To execute on-board programming, start the TMP95FY64 in Single Boot mode as follows:

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

Setting the CLK, $\overline{\text{EA}}$, and $\overline{\text{BOOT}}$ pins as shown above, and inputting a rising edge to $\overline{\text{RESET}}$ pin starts the TMP95FY64 in Single Boot Mode.

(3) Memory map

Figure 3.4(2) shows the comparison of memory maps of the single chip mode and the single boot mode.

The Internal flash memory is mapped from 10000H to 4FFFFH for the single boot mode.

Moreover, the boot ROM(Mask ROM) is mapped from FFF800H to FFFFFFH.

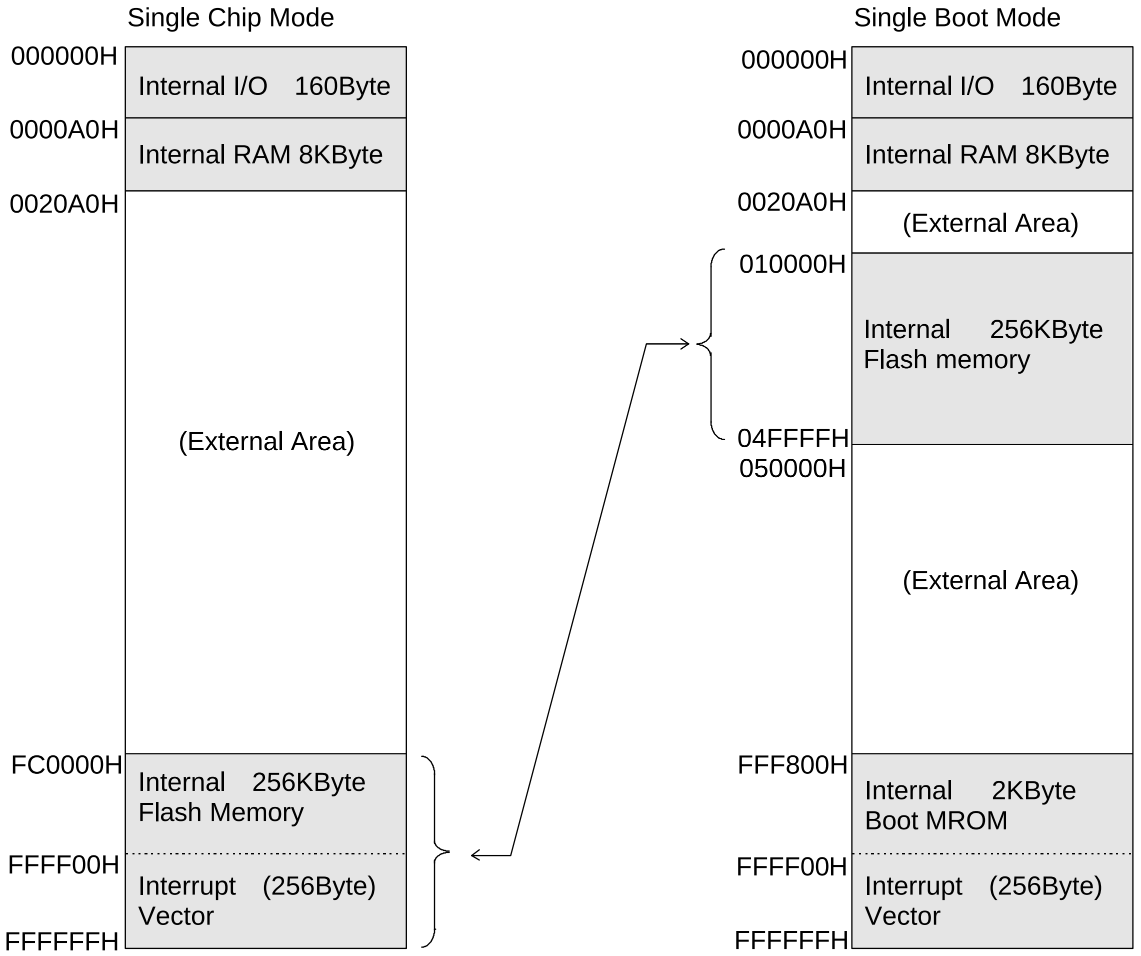


Figure 3.4(2) Comparison of Memory Maps

(4) Interface specifications

The SIO communication format in the single boot mode is shown below.

It is necessary to set the format of the communication on the writing controller side similarly to execute the on board programming.

It is possible to change as shown in Table 3.4(1) though the baud rate is initialize 9600bps.

Communication channel	: SIO channel 2
Serial transfer mode	: UART mode, full-duplex communication
Data length	: 8 bits
Parity bit	: None
Stop bit	: 1 bit
Initial Baud rate	: 9600bps

(5) Data transfer format

Table 3.4(1) to (5) show the baud rate change data, the operation command, and the data transfer format respectively.

Please read together with the following "Boot program operation explanation".

Table 3.4(1) Baud rate change data

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

Note : The AF200 supports only for 9600, 19200, 31250 and 62500 bps.

Operating frequency and baud rate in Single Boot mode. : TMP95FY64

Reference baud rate(bps)		9600		19200		31250		38400		57600		62500		76800	
Baud rate change data		28h		18h		0Ah		07h		06h		05h		04h	
Ref. Xtal(MHz)	Area(MHz)	Baud rate(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)
2.4576	2.44 to 2.48	9600	0	19200	0			38400	0						
3	2.97 to 3.03	9375	-2.34												
3.6864	3.64 to 3.74	9600	0	19200	0					57600	0				
4.9152	4.85 to 5.07	9600	0	19200	0			38400	0					76800	0
5		9766	+1.73	19531	+1.72			39063	+1.73					78125	+1.73
6	5.91 to 6.23	9375	-2.34	18750	-2.34	31250	0								
6.144		9600	0	19200	0	32000	+2.4								
7.3728	7.26 to 7.48	9600	0	19200	0			38400	0	57600	0				
8	7.84 to 8.16	9615	+0.16			31250	0					62500	0		
9.8304	9.64 to 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						
32	31.36 to 32.64	9615	+0.16	19231	+0.16	31250	0	38462	+0.16	55556	-3.55	62500	0		

Reference frequency : High speed oscillator frequency supported in Single boot mode.

When the Single boot mode is used for programming Flash memory, each of reference frequency should be used.

Area :Clock frequency area detected for reference frequency. The Single boot would not be executed at the others frequency.

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

Table 3.4(2) Operating command data

Operation command data	Operating mode
30H	Flash memory Overwrite
60H	RAM Loader
90H	Flash memory SUM

Table 3.4(3) Transfer format for Boot Program : For flash memory overwrite

	Byte number	Tool to TMP95FY64 transfer	Baud rate	TMP95FY64 to Tool transfer
BOOT ROM	Byte 1 Byte 2	Matching data(5AH) -	9600bps 9600bps	- (baud rate automatic setting) OK: Echo back data(5AH) NG: Does not send any data.
	Byte 3 Byte 4	Baud rate change data - (Table3.4(1))	9600bps 9600bps	- OK: Echo back data NG: A1H×3,A2H×3,A3H×3 *1
	Byte5 Byte 6	Operating command data (30H) -	Changed baud rate Changed baud rate	- OK: Echo back data (30H) NG: A1H×3,A2H×3,A3H×3,63H×3
	Byte 7	Changed baud rate -	Changed baud rate	OK: C1H NG: 64H×3
	Byte 8 : Byte n-2	Expanded Intel Hex format (binary) *2	Changed baud rate	-
	Byte n-1	-	Changed baud rate	OK: SUM(High) NG: Does not send any data. *3
	Byte n	-	Changed baud rate	OK: SUM(Low) NG: Does not send any data. *3
	Byte n+1	(Wait for Next Operating Command data.)	Changed baud rate	-

*1 : “xxH×3” means that after sending 3 bytes of xxH, the boot program stops operation.

*2 : See the Notes on Expanded Intel Hex Format (binary) given in Section⑥ .

*3 : See Notes on Sum given in section⑤ .

Table 3.4(4) Transfer Format for Boot Program : For RAM Loader

	Byte number	Tool to TMP95FY64 transfer data	Baud rate	TMP95FY64 to Tool transfer data
BOOT ROM	Byte 1 Byte 2	Matching data(5AH) -	9600bps 9600bps	- (baud rate automatic setting) OK : Echo back data(5AH) NG : Does not send any data.
	Byte 3 Byte 4	Baud rate change data - (Table 3.4(1))	9600bps 9600bps	- OK : Echo back data NG : A1H×3,A2H×3,A3H×3,62H×3 *1
	Byte 5 Byte 6	Operating command data (60H) -	Changed baud rate Changed baud rate	- OK : Echo back data(60H) NG : A1H×3,A2H×3,A3H×3,63H×3 *1
	Byte 7 Byte 8	Bits 23 to 16 of address store password count *2 -	Changed baud rate Changed baud rate	- OK : Does not send any data. NG : A1H×3,A2H×3,A3H×3 *1
	Byte 9 Byte 10	Bits 15 to 08 of address store password count *2 -	Changed baud rate Changed baud rate	- OK : Does not send any data. NG : A1H×3,A2H×3,A3H×3 *1
	Byte 11 Byte 12	Bits 07 to 00 of address store password count *2 -	Changed baud rate Changed baud rate	- OK : Does not send any data. NG : A1H×3,A2H×3,A3H×3 *1
	Byte 13 Byte 14	Bits 23 to 16 of password compare start address *2 -	Changed baud rate Changed baud rate	- OK : Does not send any data. NG : A1H×3,A2H×3,A3H×3 *1
	Byte 15 Byte 16	Bits 15 to 08 of password compare start address *2 -	Changed baud rate Changed baud rate	- OK : Does not send any data. NG : A1H×3,A2H×3,A3H×3 *1
	Byte 17 Byte 18	Bits 07 to 00 of password compare start address *2 -	Changed baud rate Changed baud rate	- OK : Does not send any data. NG : A1H×3,A2H×3,A3H×3 *1
	Byte 19 : Byte m	Password string *2	Changed baud rate Changed baud rate	- OK : Does not send any data. NG : A1H×3,A2H×3,A3H×3 *1
	Byte m+1 : Byte n-2	Expanded Intel Hex format(binary) *3		-
	Byte n-1	-	Changed baud rate	OK : SUM(High) NG : Does not send any data. *4
	Byte n	-	Changed baud rate	OK : SUM(Low) NG : Does not send any data. *4
RAM	-	Jump to user program start address		

*1 : “xxH×3” means that after sending 3 bytes of xxH, the boot program stops operation.

*2 : See the Notes on Password given in Section ⑦ .

*3 : See the Notes on Expanded Intel Hex Format (binary) given inSection⑥ .

*4 : See Notes on Sum given in section⑤ .

Table 3.4 (5) Transfer Format for Boot Program : For flash memory Sum

	Byte number	Tool to TMP95FY64 transfer data	Baud rate	TMP95FY64 to Tool transfer data
BOOT ROM	Byte 1 Byte 2	Matching data(5AH) -	9600bps 9600bps	- (baud rate automatic setting) OK : Echo back data(5AH) NG : Does not send any data
	Byte 3 Byte 4	Baud rate change data - (Table 3.4(1))	9600bps 9600bps	- OK : Echo back data NG : A1H×3,A2H×3,A3H×3,62H×3 *1
	Byte 5 Byte 6	Operating command data (90H) -	Changed baud rate Changed baud rate	- OK : Echo back data(90H) NG : A1H×3,A2H×3,A3H×3,63H×3 *1
	Byte 7	-	Changed baud rate	OK : SUM(High) *2 NG : -
	Byte 8	-	Changed baud rate	OK : SUM(Low) *2 NG : -
	Byte 9	(Wait for Next Operating Command data)	Changed baud rate	-

*1 : “xxH×3” means that after sending 3 bytes of xxH, the boot program stops operation.

*2 : See Notes on Sum given in section⑤ .

(6) Operation of boot program

Starting the TMP95FY64 in Single Boot mode starts the boot program. The boot program supports the following functions. For details, see the sections on① Flash Memory Overwrite command to③ Flash memory Sum command below.

1. Flash Memory Overwrite

   This command erases the flash memory in batches, up to a total of 256 KB. Then the command writes data to the specified flash memory address. The controller must send binary write data in expanded Intel hex format.
   If there is no error up to the end record, the function calculates the sum of the 256 KB flash memory and sends the result.

2. RAM Loader

   This command transfers the data sent by the controller in expanded Intel hex format to the built-in RAM. If the transfer is successfully completed, the command calculates the sum and sends the result, and starts execution of the user program. The execution start address is the address received first. This RAM Loader function enables on-board programming control unique to the user. To execute on-board programming by means of a user program, the above mentioned Flash memory command sequence must be used. (The flash memory address area must be matched with address space in the single boot mode.) Prior to execution, the RAM Loader command checks the password reference result. If the password does not match, the RAM Loader command is not executed.

3. Flash Memory Sum

   This command calculates the sum of the 256 KB flash memory and returns the result. The boot program does not support an operating command to read data from flash memory. Instead, it supports this command. Thus, reading the sum enables the user to identify the program revision.

① Flash Memory Overwrite command (Table 3.4 (3))

1. The data received in byte one are matching data. When the boot program is started in Single Boot mode, the program enters wait state for matching data. The initial baud rate of the serial channel is automatically set to 9600 bps by receiving of this matching data. The matching data are 5AH.

2. If the data received in byte one are 5AH, 5AH in byte two is sent as echo back data. If the received data are other than 5AH, the boot program sends a three-byte matching error code (61H), then stops operation.

3. The data received in byte three are baud rate change data. The seven baud rate change data are shown in Table 3.4 (1). Even if the baud rate is not changed, send the initial baud rate data (28h; 9600 bps).
   The changed baud rate is valid after echo back data are sent.

4. If the data received in byte three match one of the baud rate change data in Table 3.4 (1), the boot program uses byte four to send the received data as echo back data. Then the boot program changes the baud rate. If the data received in byte three do not match any of the baud rate change data in Table 3.4 (1), the boot program sends a three-byte baud rate change data error code (62H), then stops operation.

5. The data received in byte five are Flash Memory Overwrite command data (30H).

6. If the data received in byte five match one of the operating command data in Table 3.4 (2), the boot program uses byte six to send the received data (30H) as echo back data, then calls the Flash Memory Overwrite Processing routine. If the data received in byte five do not match any of the operating command data in Table 3.4 (2), the boot program sends a three-byte operating command error code (63H), then stops operation.

7. The data received in byte seven indicate whether batch erase (256 KB) is successfully completed.
When batch erase (256 KB) is successfully completed, the boot program sends the batch erase end code (C1H).
If an erase error occurs, the boot program sends a three-byte error code (64H). then stops operation.
On receiving the batch erase end code (C1H), the controller must send the next data.

8. The data received in byte eight to byte n-2 are regarded as binary data in expanded Intel hex format. No echo back data are sent. The Flash Memory Overwrite Processing routine ignores any data received before a start mark (3AH,":") in expanded Intel hex format. It does not send an error code while ignoring premature data. After receiving the start mark, the routine receives a record from data length to checksum. The routine sequentially writes the received write data into the specified addresses in flash memory. Bits 23 to 16 of the default address pointer are 00H. Thus, make sure to set the first record to an expanded record. After receiving a record from start mark to checksum, the routine waits for the next start mark. When a write, receive, or expanded Intel hex format error occurs, the boot program stops operation without sending an error code. After detecting the end record, the Flash Memory Overwrite Processing routine executes the Sum routine. Make sure that, after sending the end record, the controller enters wait state for receiving the sum.

9. Byte n-1 is used to send the upper byte of the sum value. Then byte n is used to send the lower byte of the sum value. For how to calculate the sum, see the Notes on Sum given in Section⑤ . The sum is calculated when only the end record is detected without a write, receive, or expanded Intel hex format error. It takes about 400ms at fc=20 MHz to calculate sum of the 256 KB flash memory area. Then the sum is sent. After sending the end record, the controller must judge whether the write to flash memory was correctly performed, depending on whether the sum value was sent.

10. When a write is performed correctly, the data received in byte n+1 are the Wait for Next Operating Command data.

② RAM Loader command (Table 3.4 (4))

1. The send/receive data in bytes one to four are the same as those for the Flash Memory Overwrite command.

2. The data received in byte five are the RAM Loader command data (60H).

3. If the data received in byte five match one of the operating command data in Table 3.4 (2), byte six is used to send the echo back of the received data (60H). Then the RAM Loader processing routine is called. If the data received in byte five do not match any of the operating command data in Table 3.4 (2), the boot program sends a three-byte operating command error code (63H), then stops operation.

4. The data received in byte seven are those in bits 23 to 16 of the address used to store the password.
Three bytes of the address are needed for storing the password count. The data at this address are used as the password count. Note that if the password count is eight or less, execution of the RAM Loader command is canceled.

5. If there is no receive error in the data received in byte seven, the processing routine does not send any data using byte eight. If there is a receive error, the processing routine sends three bytes of the corresponding error code, then stops operation.

6. Bytes nine to 12 are used to receive data from bits 15 to 8 and from bits 7 to 0 of the address used to store the password. These bytes are also used to send an error code for receive error, if any. For this operation, see 4 and 5 above.

7. The data received in byte 13 are those of bits 23 to 16 of the password compare start address. Three bytes of the password compare start address are required. The password comparison starts from this address.

8. If there is no receive error in the data received in byte 13, the processing routine does not send any data using byte 14. If there is a receive error, the processing routine sends three bytes of the corresponding error code, then stops operation.

9. Bytes 15 to 18 are used to receive and send data from bits 15 to 8 and from bits 7 to 0 of the password compare start address. For this operation, see 7 and 8 above.

10. Bytes 19 to m are used to receive password data. The password count is data (N) indicated by the address used to store the password count. N passwords are compared from the password compare start address.
The controller must send the N-byte password data. If the password does not match, the processing routine stops operation without sending an error code.

11. The data received in bytes m+1 to n-2 are regarded as binary data in expanded Intel hex format. No echo back data are sent. The RAM Loader Processing routine ignores any data received before a start mark (3AH,,":") in expanded Intel hex format. It does not send an error code while ignoring premature data. After receiving the start mark, the routine receives a record from data length to checksum. The routine sequentially writes the received write data into the specified addresses in RAM. Bits 23 to 16 of the default address pointer are 00H. Thus, the first record is not necessarily an expansion record. After receiving a record from the start mark to checksum, the routine waits for the next start mark. When a receive or expanded Intel hex format error occurs, the routine stops operation without sending the error code. After detecting the end record, the RAM Loader processing routine executes the sum routine. Make sure that, after sending the end record, the controller enters wait state for receiving the sum.

12. Byte n-1 is used to send the upper byte of the sum value. Then byte n is used to send the lower byte of the sum value. For the method of calculating the sum, see the Notes on Sum given in Section⑤ . The sum is calculated when only the end record is detected without a receive or expanded Intel hex format error. The time required for calculating the sum is almost proportional to the number of data items which have been written into RAM. For example, the time required to calculate a 4K RAM area at fc=20 MHz is approximately 6 ms. Then the sum is sent. After sending the end record, the controller must judge whether the write to RAM was correctly performed, depending on whether the sum value was sent.

13. When the sum has been sent, the boot program jumps to the address of the first data stored in the RAM area in expanded Intel hex format.

③ Flash Memory Sum command (Table 3.4 (5))

1. The send/receive data in bytes one to four are the same as those for the Flash Memory Overwrite command.

2. The data received in byte five are the Flash Memory Sum command data (90H).

3. If the data received in byte five match one of the operating command data in Table 3.4 (2), byte six is used to send the echo back of the received data (90H). Then the Flash Memory Sum processing routine is called. If the data received in byte five do not match any of the operating command data in Table 3.4 (2), the boot program sends a three-byte operating command error code (63H), then stops operation.

4. Byte seven is used to send the upper byte of the sum value. Then byte eight is used to send the lower byte of the sum value. For the method of calculating the sum, see the Notes on Sum given in Section⑤ .

5. The data received in byte 9 are the Wait for Next Operating Command data.

④ Boot program send data

The boot program sends the processing status in code form to the controller. Listed below are the send data (processing codes):

Table 3.4(6) Boot Program Send Data

send data	send data description
C1H	Normally end of Chip erasing.
62H,62H,62H	Baud rate change error is occurred.
63H,63H,63H	Operating command error is occurred.
64H,64H,64H	Erasing error is occurred.
A1H,A1H,A1H	Framing error in receive data is occurred. *1
A2H,A2H,A2H	Parity error in receive data is occurred. *1
A3H,A3H,A3H	Overrun error in receive data is occurred. *1

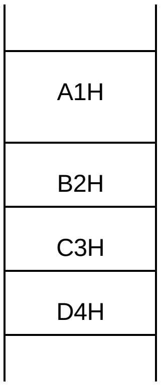
*1 : If this error is generated while data in expanded Intel Hex format are being received, the receive error code is not sent.

⑤ Sum calculation

1. Calculation method

The Flash Memory Sum command returns, in words, the result of summing in bytes. That is, data are read in bytes, and calculated, but the result is returned in words.

Example



If the four bytes on the left are the target data, the sum is determined as follows:

A1H+B2H+C3H+D4H=02EAH

SUM(HIGH) =02H
SUM(LOW) =EAH

The above calculation method is used to obtain the sums returned when the Flash Memory Overwrite command, RAM Loader command, and Flash Memory Sum command are executed.

2. Sum Target Data

Table 3.4 (7) shows the sum target data.

Table 3.4(7) Sum Target Data

Operating mode	Calculation target data	Remarks
Flash memory overwrite command	Data written in entire flash memory (256KB)	Sum target data are not write data received in flash memory or RAM. Even if the received addresses are not consecutive and there are some areas without any data written, the data which have been written are the target.
RAM Loader command	Data written starting at address received first, going on to address received last.	
Flash memory Sum command	Data written to entire flash memory (256KB)	–

⑥ Notes on expanded Intel hex format (binary)

1. For the Flash Memory Overwrite command, the first record must be an expansion record, because the flash memory of the TMP95FY64 is allocated to addresses starting from 10000H, thus bits 23 to 16 of the default address pointer are 00H.

2. For the RAM Loader command, the first record does not necessarily have to be an expansion record, because bits 23 to 16 of the default address pointer are 00H.

3. After the checksum of a record is received, the program enters wait state for the next record start mark (3AH, ":"). Thus, data other than 3AH are ignored.

4. Make sure that after sending the checksum of the end record, the controller program does not do anything other than wait for the 2-byte receive data (upper and lower data of the sum). This is because after receiving the checksum of the end record, the boot program calculates the sum and returns the result in two bytes.

5. When a write (for Flash Memory Overwrite command only), receive, or expanded Intel hex format error occurs, the boot program stops operation without sending the error code. Expanded Intel hex format errors occur in the following cases:

- When TYPE is other than 00H, 01H, or 02H
- When a checksum error occurs
- When data length of an expansion record (TYPE=02H) is other than 02H
- When the address of an expansion record (TYPE=02H) is other than 0000H
- When byte two data of an expansion record (TYPE=02H) are other than 00H
- When data length of an end record (TYPE=01H) is other than 00H
- When the address of an end record (TYPE=01H) is other than 0000H

Example : When data are written to the area between addresses 1FFF8H and 2002FH, the operating command data are as shown below:

Table 3.4(8) Example of Transfer Format for Flash memory overwrite command data

Data transfer direction	Data description Expanded Intel Hex format (byte 8 and byte n-2 in Table3.4(3))	Data
Controller to TMP95FY64	Expansion record	: 02 0000 02 1000 EC zz
Controller to TMP95FY64	Data record(data length : 08H)	: 08 FFF8 00 xxxxxx CS zz
Controller to TMP95FY64	Expansion record	: 02 0000 02 2000 DC zz
Controller to TMP95FY64	Data record(data length : 30H)	: 30 0000 00 yyyyyyyy CS zz
Controller to TMP95FY64	End record	: 00 0000 01 FF ww
TMP95FY64 to Controller	SUM(upper) (byte n-1 in Table 3.4(3))	SUM(High)
TMP95FY64 to Controller	SUM(lower) (byte n in Table 3.4(3))	SUM(Low)
Controller to TMP95FY64	Operating command (byte n+1 in Table 3.4(3))	Next operating command data

Notes :	“:”	: 3AH(start mark)
	xx, yy	: Data to be written to flash memory
	CS,EC,DC,FF	: check sum data
	zz	: Controller does not have to send.
	ww	: Controller must not send.

⑦ Notes on passwords

A password cannot be specified for the entire flash memory area (256 KB). It can only be specified for a limited area, which is bounded by the addresses from 12000H to 4DFFFH. Figure 3.4 (3) shows a schematic representation of the password area.

1. Address to store password count (PNSA)

The contents at the address specified by PNSA is the password count (N). In the following case, a password error occurs.

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

2. Password Compare Start address (PCSA)

Password comparison starts from the address specified by PCSA. The specified password area is from PCSA to PCSA+N. In the following case, a password error occurs.

- PCSA < address 12000H
- Address 2DFFFH < PCSA+N-1
- If the same data continues for three bytes or more in the specified password area:
  If the data in the vector area (4FF00H to 4FFFFH) are all FFH, the area is regarded as unprogrammed, thus checking is not performed.

3. Password string

The password string is compared with the data in flash memory.

- A password error occurs when the received data do not match the data in flash memory.

4. Password error processing

When a password error occurs, the program stops operation.

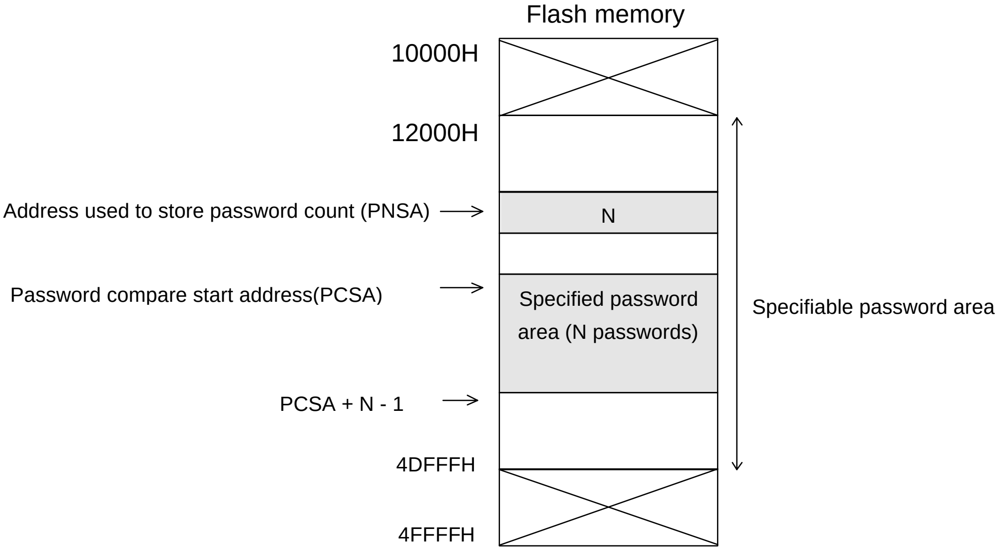
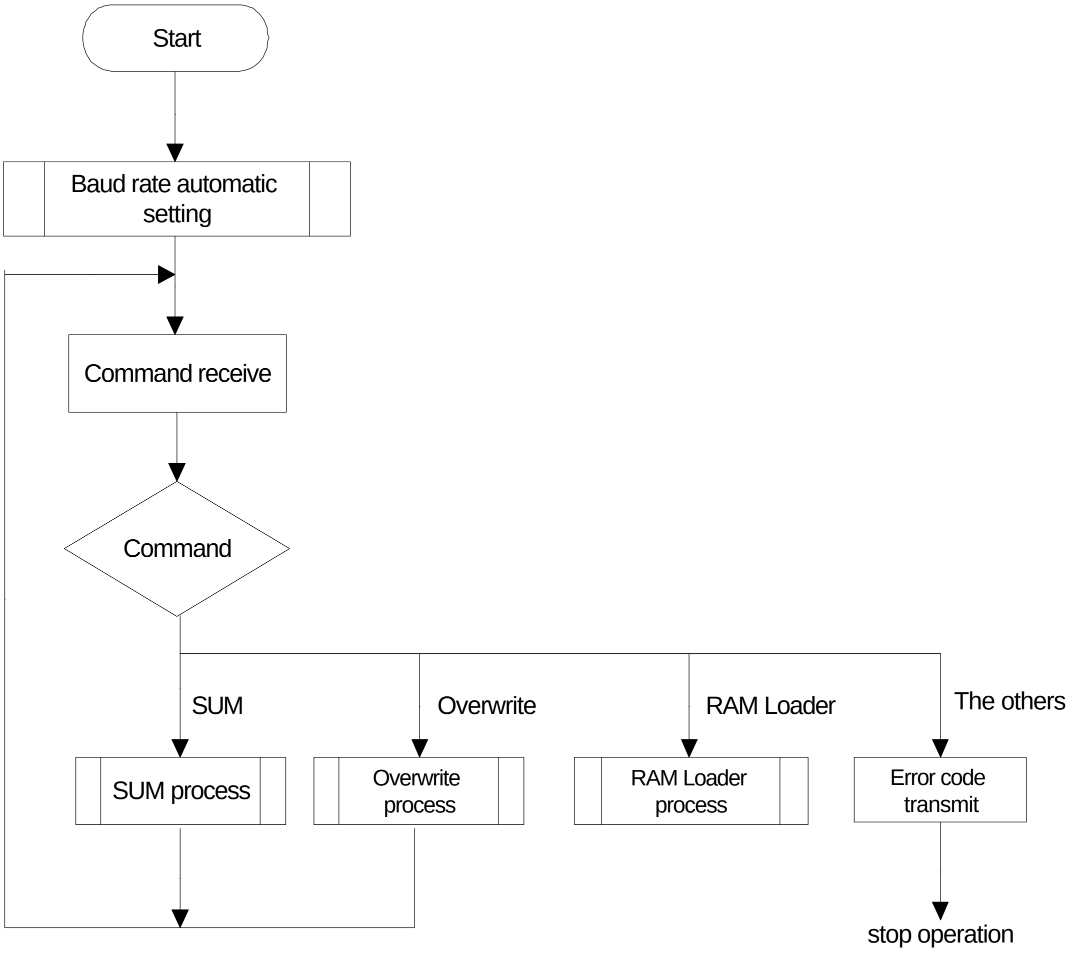
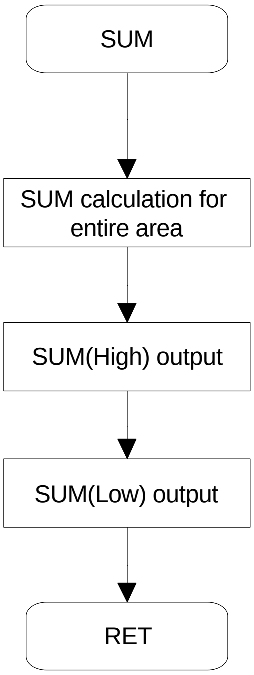

Flash memory
10000H
12000H
Address used to store password count (PNSA)
N
Password compare start address(PCSA)
Specified password area (N passwords)
PCSA + N - 1
4DFFFH
4FFFFH
Specifiable password area

Figure 3.4(3) Schematic Representation of Password Area

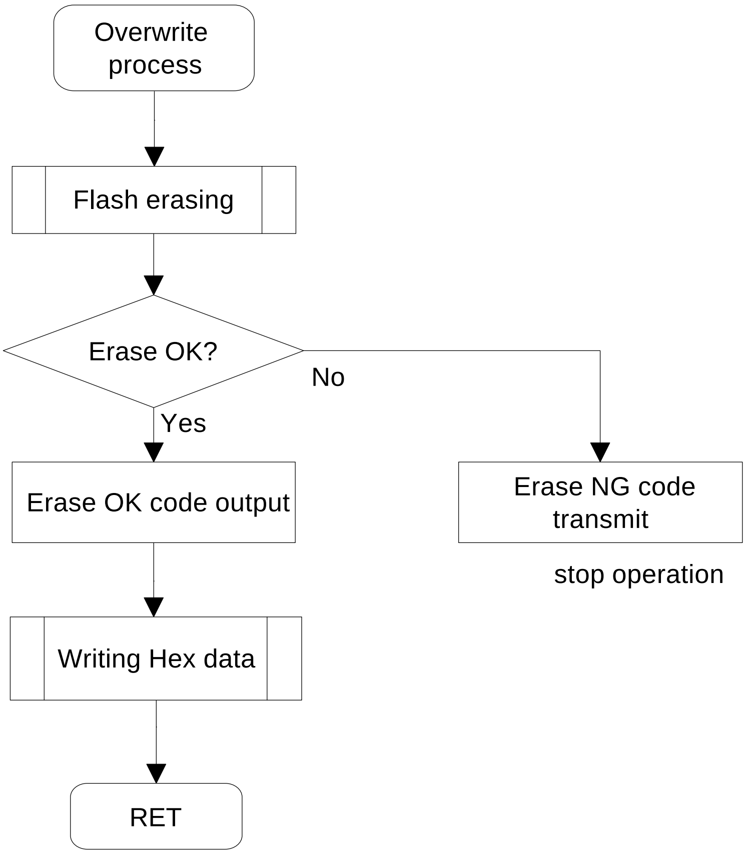
Single Boot General Flow



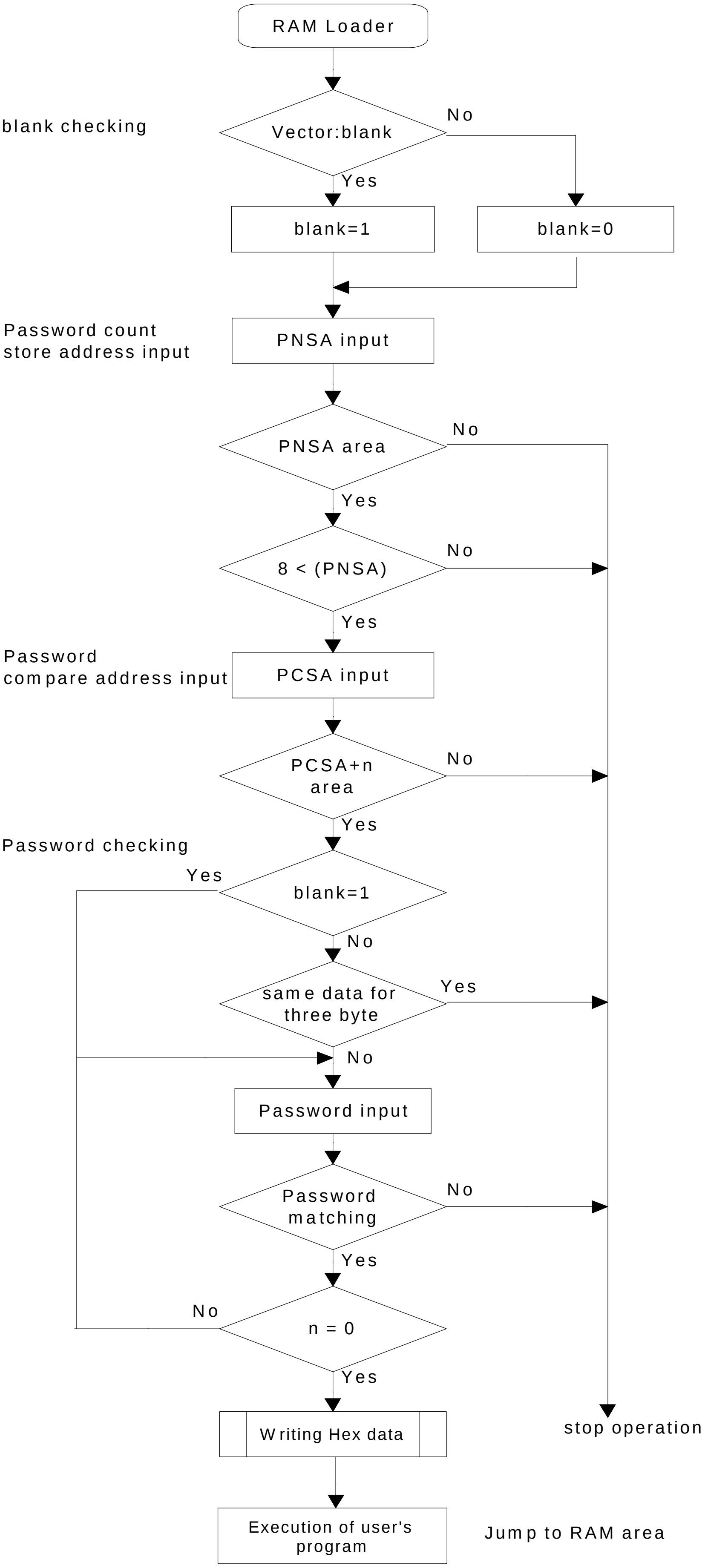
(1) SUM command



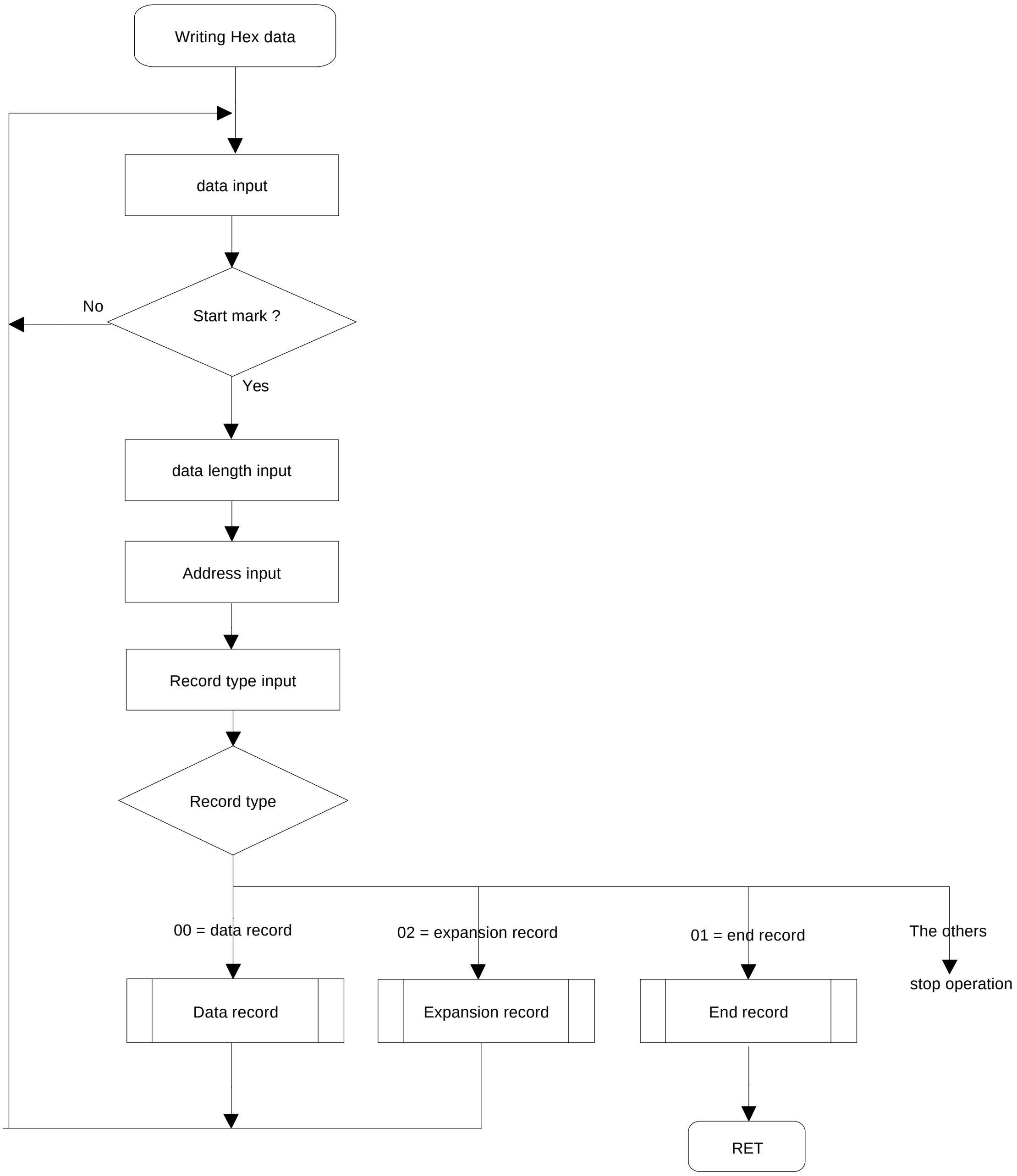
(2) Overwrite command



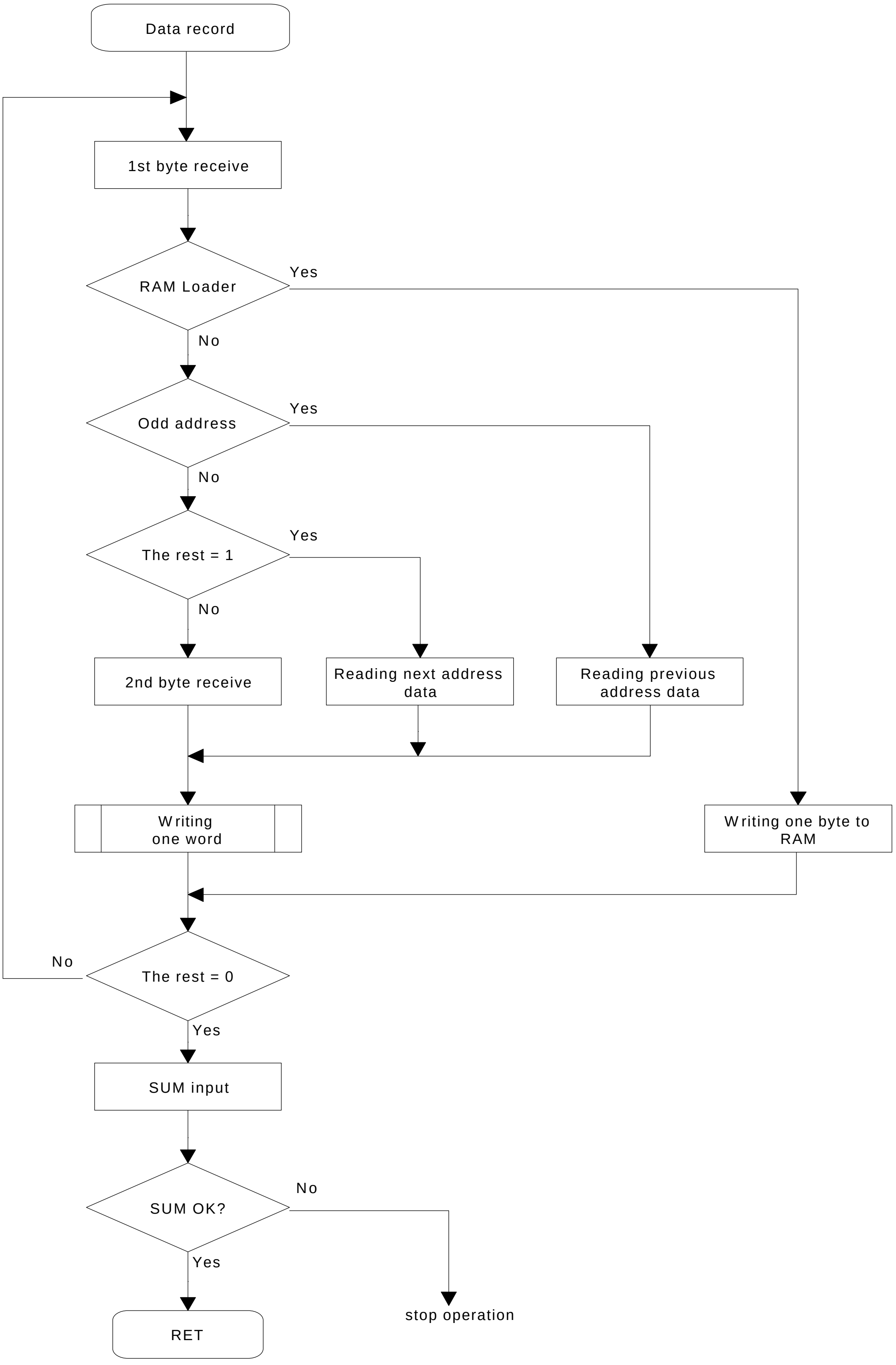
(3) RAM Loader command



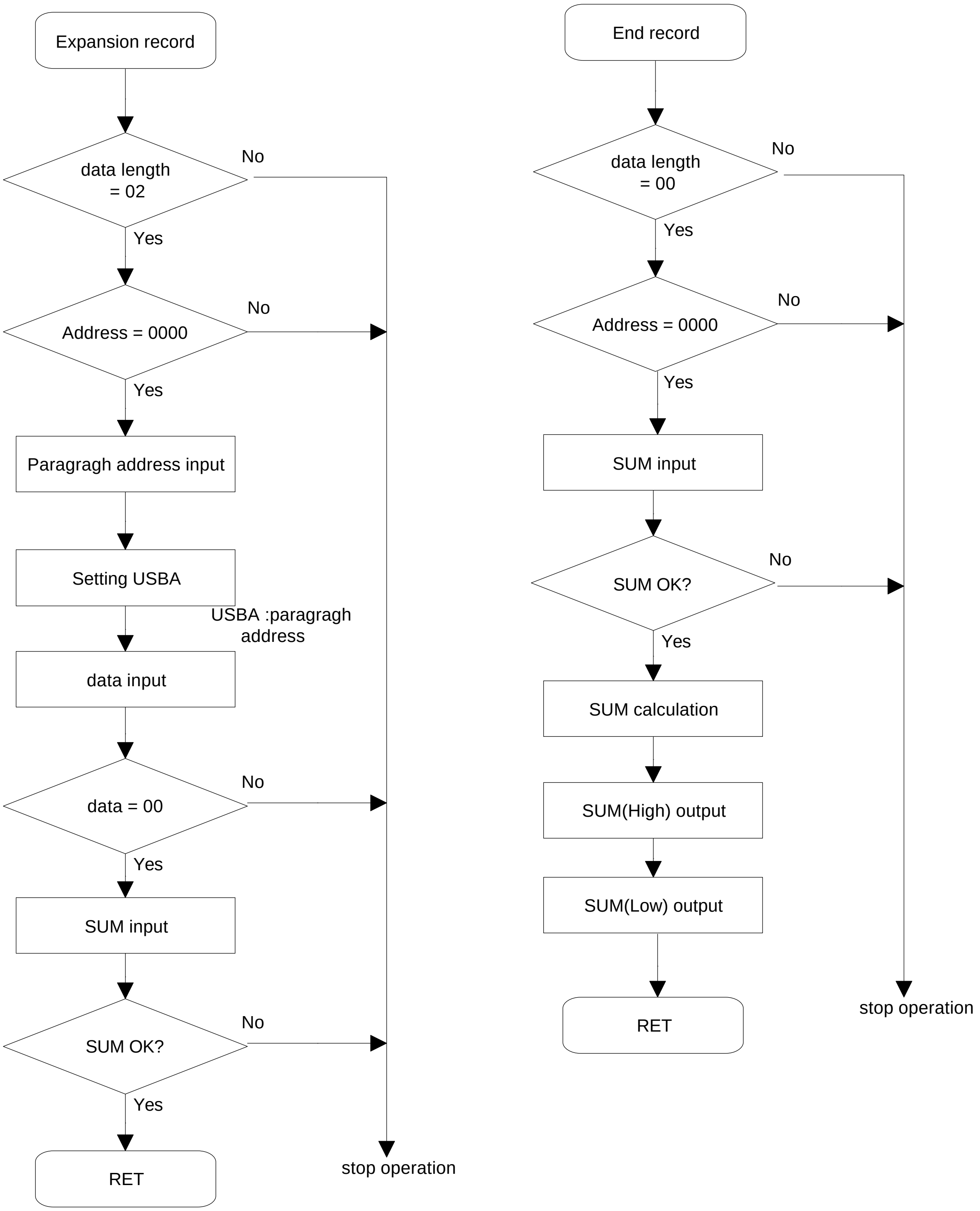
(2) - 1 Writing HEX data



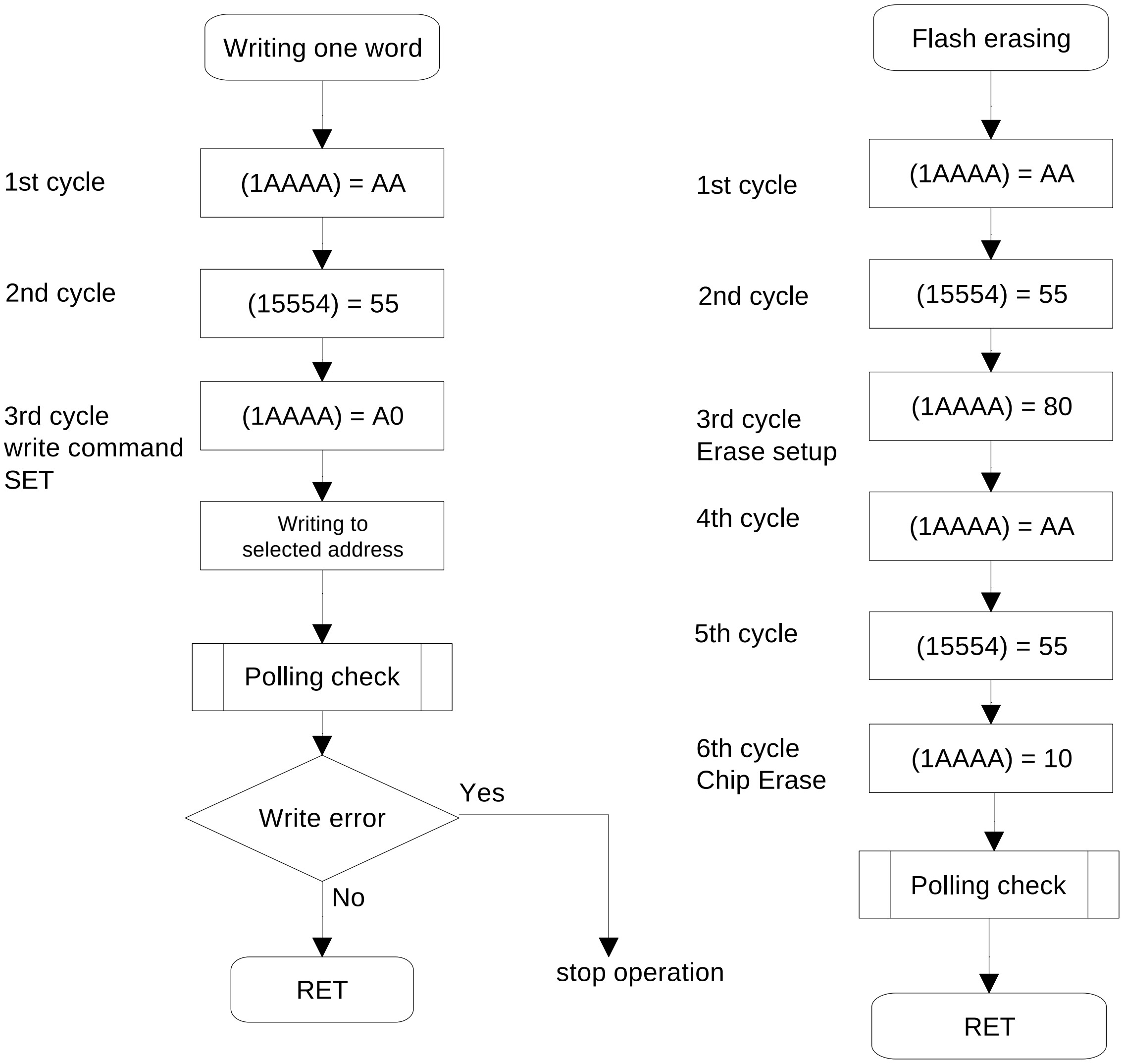
(2) - 1 - 1 Data record



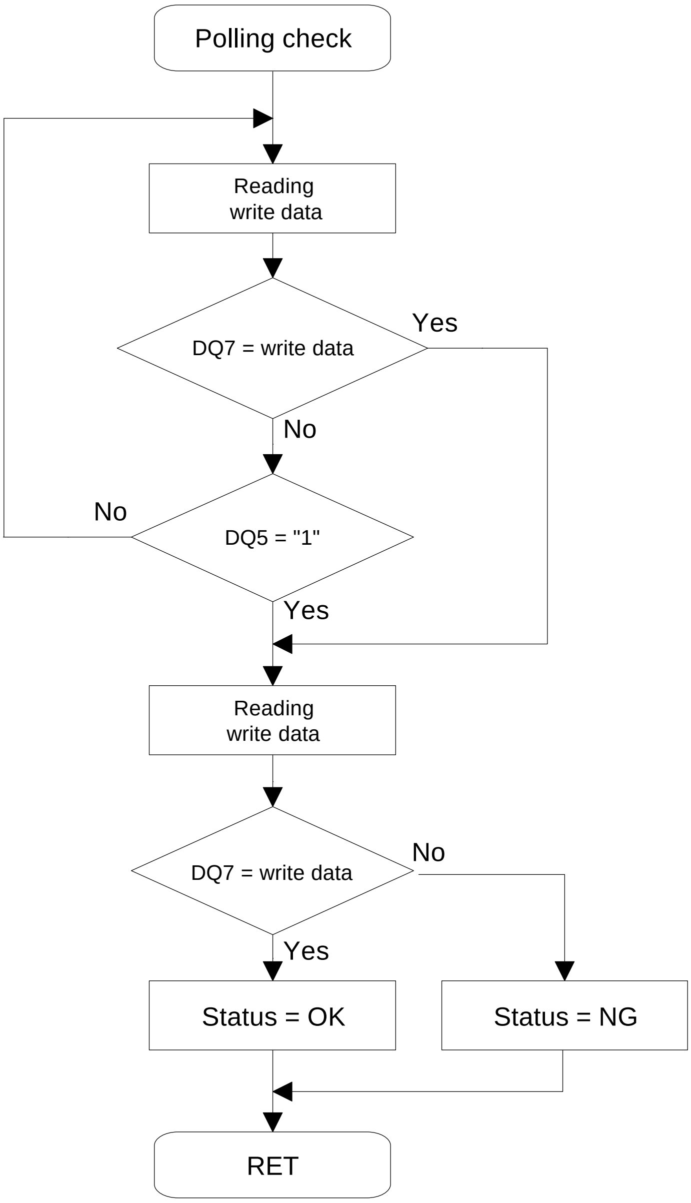
(2) - 1 - 2 Expansion record
(2) - 1 - 3 End record



(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

Symbol	Parameter	Rating	Unit
V_{cc}	Power Supply Voltage	− 0.5 to + 6.5	V
V_{IN}	Input Voltage	− 0.5 to Vcc + 0.5	V
ΣI_{OL}	Output current (total)	+ 120	mA
ΣI_{OH}	Output current (total)	− 120	mA
P_D	Power Dissipation (Ta = + 70 °C)	600	mW
T_{SOLDER}	Soldering Temperature (10 s)	+ 260	°C
T_{STG}	Storage Temperature	− 65 to + 150	°C
T_{OPR}	Operating Temperature	− 20 to + 70	°C
N_{EW}	Erase/Program Cycle Capability	1,000	Cycle

Note : Exceeding the maximum ratings for the LSI can cause permanent damage.

4.2 DC Electrical Characteristics

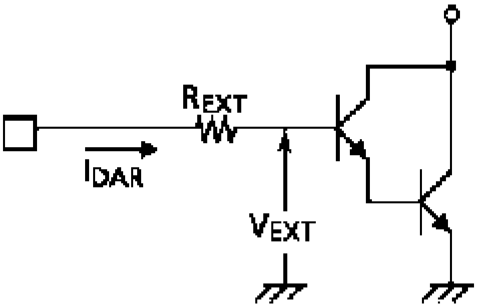
(1) Vcc = + 5 V ± 10%, Ta = − 20 to + 70 °C (fc = 8 to 25 MHz)

(Typical values are for Ta = + 25°C, VCC = + 5V.)

Symbol	Parameter	Min	Max	Unit	Test Condition
V_{IL}	Input Low Voltage (D0 to 15)	− 0.3	0.8	V	
V_{IL1}	Port 2 to A (except P56, P70, P72, P73, P75)	− 0.3	0.3 Vcc	V	
V_{IL2}	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0 to 4	− 0.3	0.25 Vcc	V	
V_{IL3}	$\overline{\text{EA}}$, AM8/16	− 0.3	0.3	V	
V_{IL4}	X1	− 0.3	0.2Vcc	V	
V_{IH}	Input High Voltage (D0 to 15)	2.2	Vcc + 0.3	V	
V_{IH1}	Port 2 to A (except P56, P70, P72, P73, P75)	0.7 Vcc	Vcc + 0.3	V	
V_{IH2}	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0 to 4	0.75 Vcc	Vcc + 0.3	V	
V_{IH3}	$\overline{\text{EA}}$, AM8/16	Vcc − 0.3	Vcc + 0.3	V	
V_{IH4}	X1	0.8Vcc	Vcc + 0.3	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = − 400 μA
V_{OH1}		0.75 Vcc		V	I_{OH} = − 100 μA
V_{OH2}		0.9 Vcc		V	I_{OH} = − 20 μA
I_{DAR}	Darlington Drive Current (8 Output Pins max.)	− 1.0	− 3.5	mA	V_{EXT} = 1.5 V R_{EXT} = 1.1 kΩ
I_{LI}	Input Leakage Current	0.02 (Typ)	± 5	μA	0.0 ≤ Vin ≤ Vcc
I_{LO}	Output Leakage Current	0.05 (Typ)	± 10	μA	0.2 ≤ Vin ≤ Vcc − 0.2
I_{cc}	Operating Current (RUN)	40 (Typ)	100	mA	fc = 25 MHz
	IDLE2	30 (Typ)	40	mA	
	IDLE1	3.5 (Typ)	10	mA	
	STOP (Ta = − 20 to + 70 °C)	0.5 (Typ)	150	μA	0.2 ≤ Vin ≤ Vcc − 0.2
V_{STOP}	Power Down Voltage (@STOP, RAM Back up)	2.0	6.0	V	V_{IL2} = 0.2 Vcc, V_{IH2} = 0.8 Vcc
R_{RP}	Pull Up Registance	45	160	kΩ	
C_{IO}	Pin Capacitance		10	pF	fc = 1 MHz
V_{TH}	Schmitt Width $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0 to 4	0.4	1.0 (Typ)	V	

Note : I_{DAR} guarantees up to eight pins from any output port.

Refer: I_{DAR} definition diagram.



4.3 AC Electrical Characteristics

(1) Vcc = +5 V ± 10 %, Ta = −20 to +70 °C

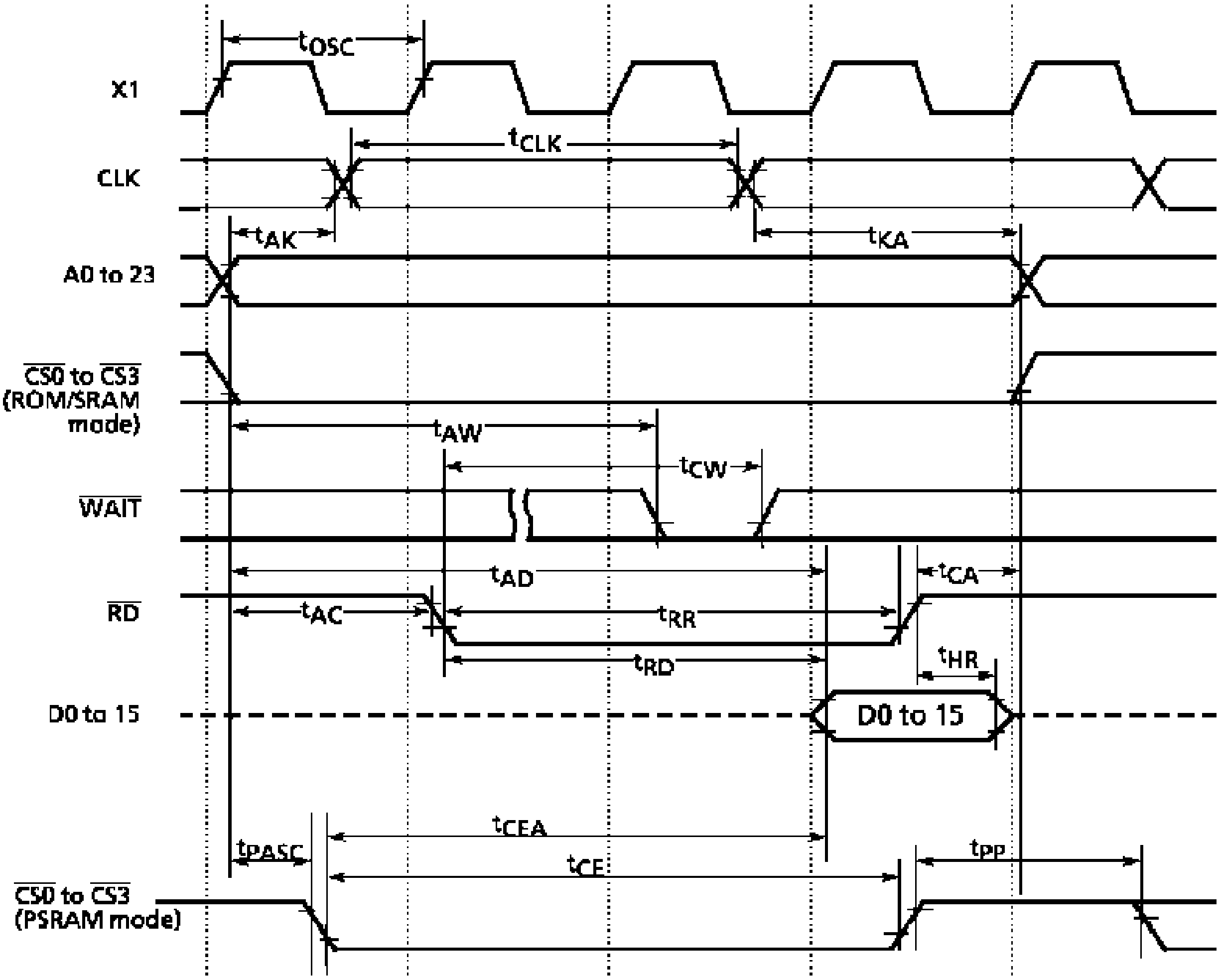
(fc = 8 MHz to 25 MHz)

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

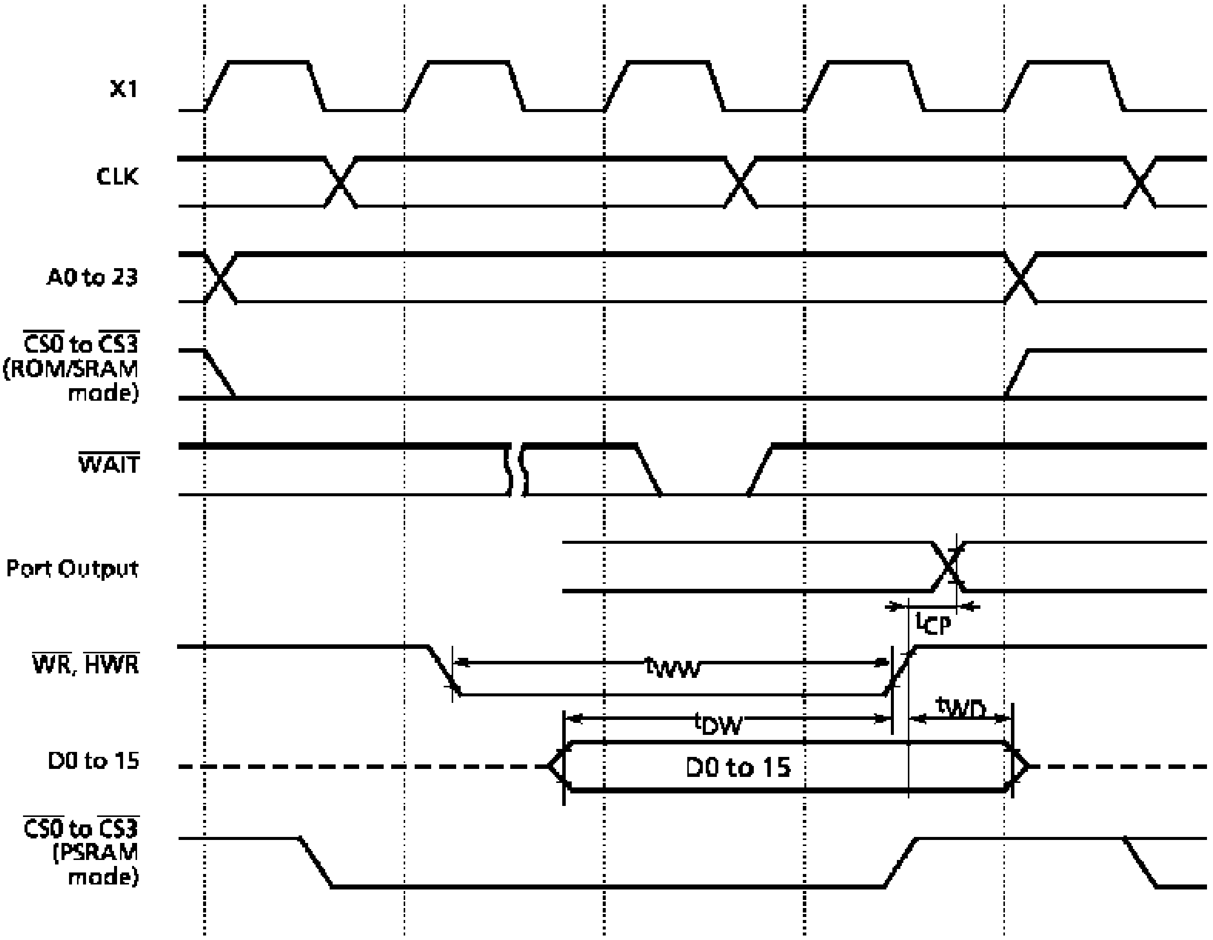
AC measuring conditions

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

(3) Read Cycle


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

(4) Write Cycle


X1
CLK
A0 to 23
$\overline{CS0}$ to $\overline{CS3}$ (ROM/SRAM mode)
$\overline{WAIT}$
Port Output
t_{CP}
$\overline{WR}$, $\overline{HWR}$
t_{WW}
t_{DW}
t_{WD}
D0 to 15
D0 to 15
$\overline{CS0}$ to $\overline{CS3}$ (PSRAM mode)

4.4 Serial Channel Timing

(1) I/O interface mode

① SCLK input mode

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

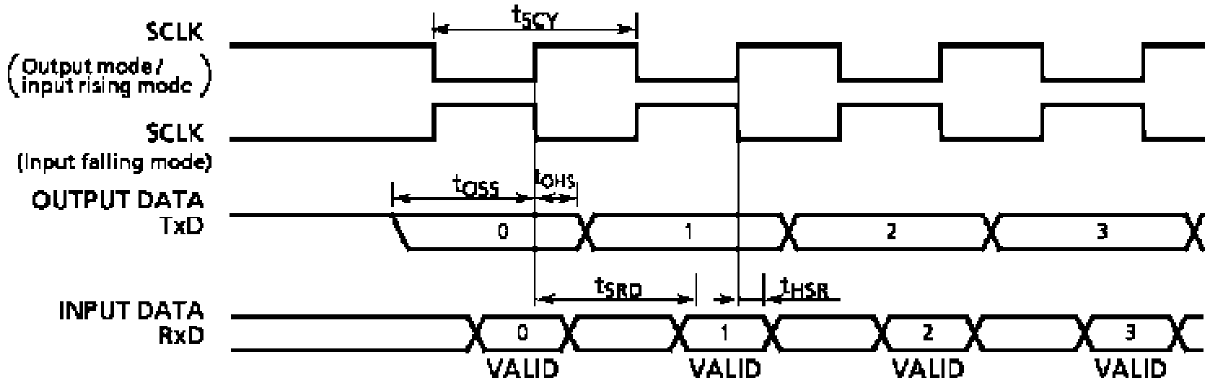
Symbol	Parameter	Formula		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle	16x		1.6		0.64		μs
t_{OSS}	Output Data → SCLK rise/fall*	$t_{SCY}/2 - 5x - 50$		250		70		ns
t_{OHS}	SCLK rise/fall* → Output Data hold	5x − 100		400		100		ns
t_{HSR}	SCLK rise/fall* → input data hold	0		0		0		ns
t_{SRD}	SCLK rise/fall* → valid data input		$t_{SCY} - 5x - 100$		1000		340	ns

*) SCLK rise/fall: In SCLK rising edge mode, SCLK rising edge timing; in SCLK falling edge mode, SCLK falling edge timing

② SCLK output mode

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

Symbol	Parameter	Formula		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle (programmable)	16x	8192x	1.6	819.2	0.64	327.6	μs
t_{OSS}	Output Data → SCLK rising edge	$t_{SCY} - 2x - 150$		1250		410		ns
t_{OHS}	SCLK rising edge → Output Data hold	2x − 80		120		0		ns
t_{HSR}	SCLK rising edge → Input Data hold	0		0		0		ns
t_{SRD}	SCLK rising edge → valid data input		$t_{SCY} - 2x - 150$		1250		410	ns



(2) UART Mode (SCLK0 to 2 External Input)

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

Symbol	Parameter	Formula		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle	4x + 20		420		180		ns
t_{SCYL}	Low-level SCLK pulse width	2x + 5		205		85		ns
t_{SCYH}	High-level SCLK pulse width	2x + 5		205		85		ns

4.5 A/D Conversion Characteristics

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

Symbol	Parameter		Min	Typ	Max	Unit	Test Conditions
V_{REFH}	A/D analog reference supply voltage (+)		Vcc − 0.2		Vcc	V	
V_{REFL}	A/D analog reference supply voltage (−)		Vss		Vss + 0.2		
AV_{CC}	Analog reference voltage		Vcc − 0.2		Vcc		
AV_{SS}	Analog reference voltage		Vss		Vss + 0.2		
V_{AIN}	Analog input voltage		V_{REFL}		V_{REFH}		
I_{REF}	Analog reference voltage supply current	<VREFON> = 1			3.7	mA	Vcc = 5 V ± 10 %
		<VREFON> = 0		0.02	5.0	μA	Vcc = 5 V ± 10 %
E_T	Total tolerance			± 1	± 3	LSB	Vcc = 5 V ± 10 %
	(excludes quantization error)						

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

Note 2 : Power supply current ICC from the VCC pin includes the power supply current from the AVCC pin.

4.6 D/A Conversion Characteristics

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

Symbol	Parameter	Min	Typ	Max	Unit	Condition
AV_{CC}	Analog reference voltage	Vcc − 0.2		Vcc	V	
AV_{SS}	Analog reference voltage	V_{SS}		Vss + 0.2		
	Total tolerance			7.0 4.0 3.5	LSB LSB LSB	R = 1 MΩ (Note) R = 5 MΩ (Note) R = 10 MΩ (Note)
	Differential linear error		2.0		LSB	

Note : R is the external load resistance on the D/A converter output pin (DAOUT0, DAOUT1).

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

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

Symbol	Parameter	Calculator		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{VCK}	External input clock cycle	8x + 100		900		420		ns
t_{VCKL}	External low-level input clock pulse width	4x + 40		440		200		ns
t_{VCKH}	External high-level input clock pulse width	4x + 40		440		200		ns

4.8 Interrupt Operation

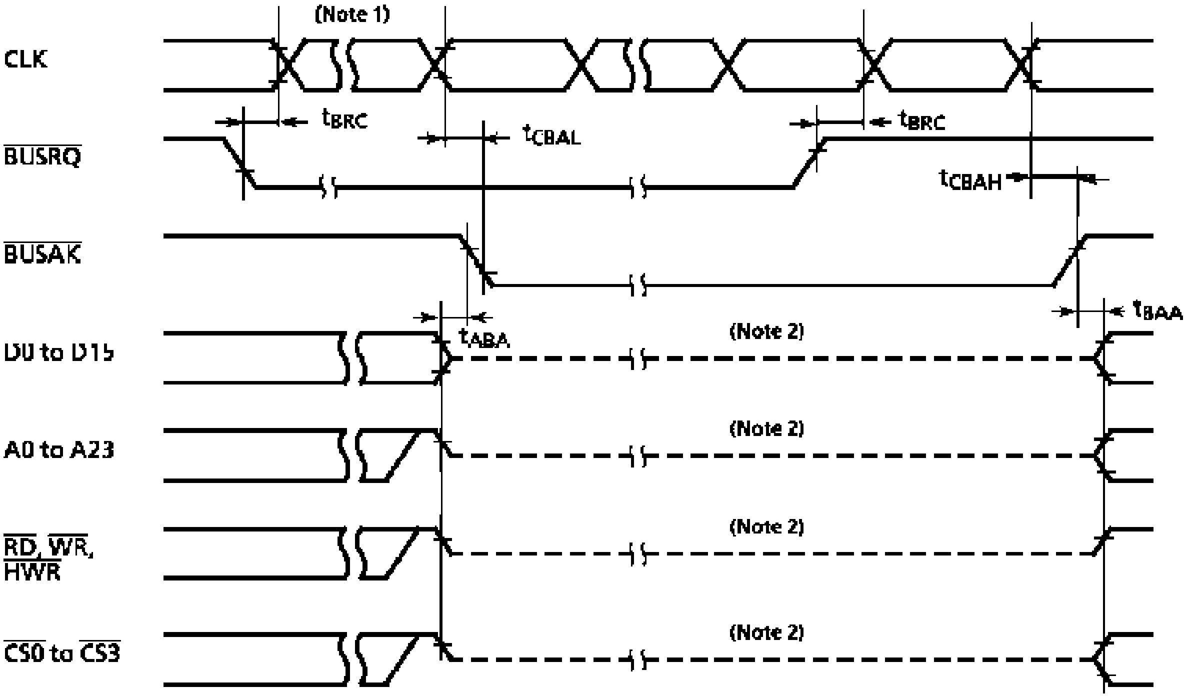
Vcc = +5V ± 10 %, Ta = −20 to +70 °C (fc = 8 to 25 MHz)

Symbol	Parameter	Calculator		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	$\overline{\text{NMI}}$, INT0 to 4 low-level pulse width	4x		400		160		ns
t_{INTAH}	$\overline{\text{NMI}}$, INT0 to 4 high-level pulse width	4x		400		160		ns
t_{INTBL}	INT5 to INT8 low-level pulse width	8x + 100		900		420		ns
t_{INTBH}	INT5 to INT8 high-level pulse width	8x + 100		900		420		ns

4.9 Bus Request/Bus Acknowledge Timing

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

Symbol	Parameter	Calculator		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{BRC}	$\overline{\text{BUSRQ}}$ setup time for CLK	120		120		120		ns
t_{CBAL}	CLK→$\overline{\text{BUSAK}}$ fall		2.0x + 120		320		200	ns
t_{CBAH}	CLK→BUSAK rise		0.5x + 40		90		60	ns
t_{ABA}	Time from output buffer off until $\overline{\text{BUSAK}}$ falling edge	0	80	0	80	0	80	ns
t_{BAA}	Time from BUSAK rising edge until output buffer on	0	80	0	80	0	80	ns



Note 1 : When $\overline{\text{BUSRQ}}$ goes to low level to request bus release, if the current bus cycle is yet complete due to a wait, the bus is not released until the wait completes.

Note 2 : The dotted line indicates only that the output buffer is off, not that the signal is at middle level. Immediately after bus release, the signal level prior to the bus release is held dynamically by the external load capacitance. Therefore, designs should allow for the fact that when using an external resistor or similar to fix the signal level while the bus is released, after bus release a delay occurs before the signal goes to its fixed level (due to the CR time constant). The internal programmable pull-up resistor continues to function in accordance with the internal signal level.