TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/L1 Series

TMP91FY28

TOSHIBA CORPORATION

Semiconductor Company

Preface

Thank you very much for making use of Toshiba microcomputer LSIs.
Before use this LSI, refer the section, “Points of Note and Restrictions”.

CMOS 16-Bit Microcontroller

TMP91FY28FG

1. Outline

The TMP91FY28 is a high-speed and high-performance 16-bit microcontroller suitable for low-voltage, low-power applications.

The TMP91FY28FG comes in a 100-pin mini flat package. Features of the TMP91FY28FG include the following:

(1) High-speed 16-bit CPU (900/L1 CPU)
- Instruction set is upwardly assembly-code compatible.
- 16-Mbyte linear address space
- Architecture based on general-purpose registers and register banks
- 16-bit multiply/divide instructions and bit transfer/arithmetic instructions
- 4-channel Micro DMA (1.6 μs/2 bytes at 10 MHz)

(2) Minimum instruction execution time: 400 ns (at 10 MHz)

(3) 8-Kbyte on-chip RAM
256-Kbyte on-chip flash
2-Kbyte masked ROM that contains software bootstrap

(4) External memory expansion
- 16-Mbyte off-chip address space for code and data
- External bus interface with dynamic bus sizing for 8-bit and 16-bit data ports

(5) 4-channel 8-bit timer

(6) 2-channel 16-bit timer

(7) 1-channel general-purpose serial interface
- Both UART and synchronous transfer modes are supported.

(8) 2-channel serial bus interface
- Either I²C mode or clock-synchronous mode can be selected.

(9) 8-channel 10-bit AD converter (with internal sample/hold)

(10) Watchdog timer

(11) Key wakeup interrupt with 8-bit inputs

(12) WAKE output pin

(13) BCD adder/subtractor

(14) Program patch logic

- 6 banks of registers

(15) 4-channel chip select/wait controller

(16) 48 interrupt sources

- 9 CPU interrupts: Triggered by software interrupt instruction or upon the execution of an undefined instruction
- 21 internal interrupts: 7 priority levels
- 18 external interrupts: 7 priority levels (16 interrupts supporting selection of triggering edge)

(17) 80-pin input/output ports

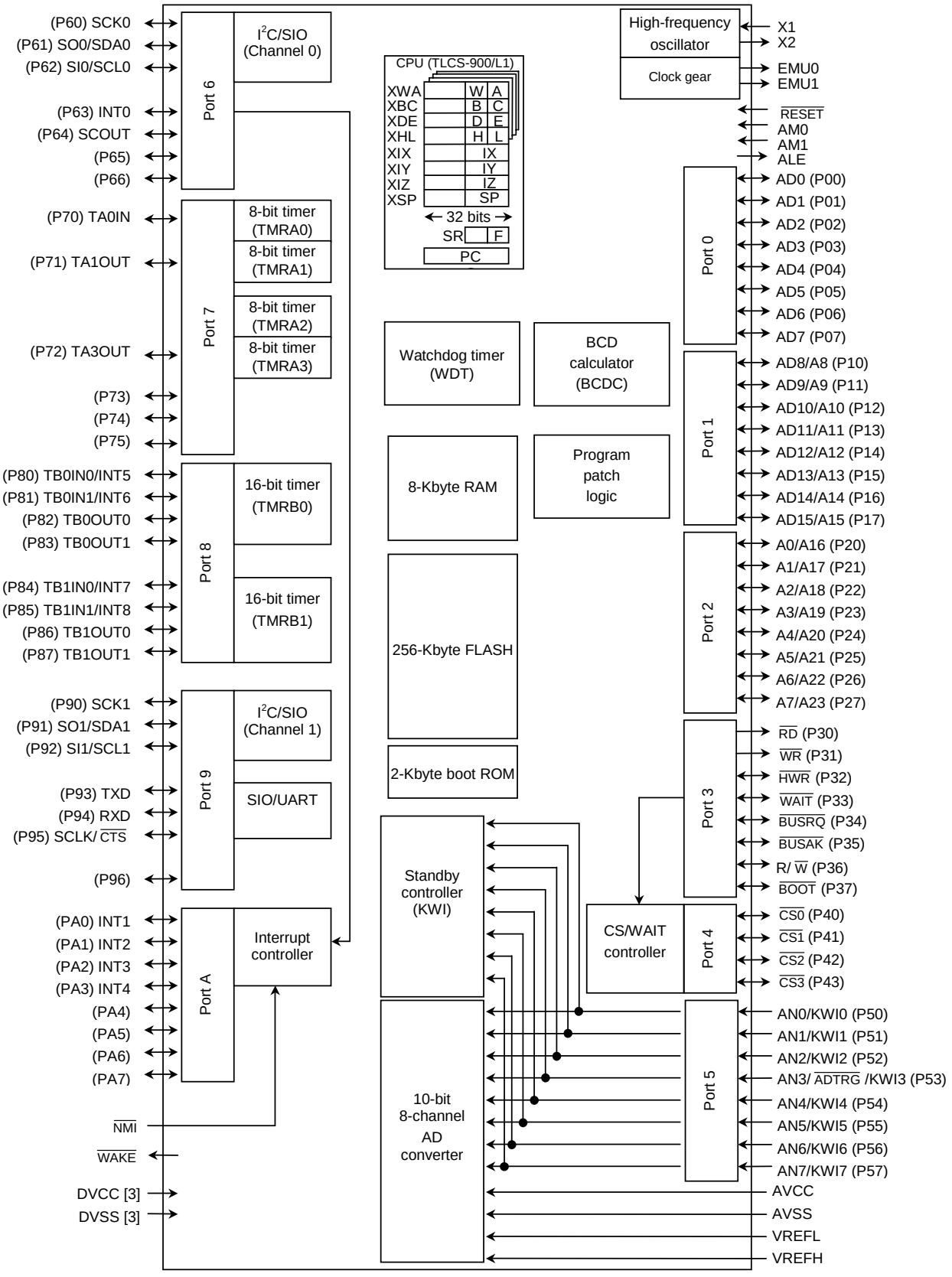
(18) Three HALT modes: Programmable IDLE2, IDLE1 and STOP

(19) Clock control

- Clock gear: Switches the frequency of high-frequency clock within the range from fc to fc/16

(20) Operating voltage range: VCC = 1.8 to 2.6 V (fc max = 10 MHz)

(21) Package: P-LQFP100-1414-0.50F



( ): Initial pin function after reset

Figure 1.1 TMP91FY28 Block Diagram

2. Signal Description

This section contains pin assignments for the TMP91FY28 as well as brief descriptions of the TMP91FY28 input and output signals.

2.1 Pin Assignment

The following illustrates the TMP91FY28FG pin assignment.

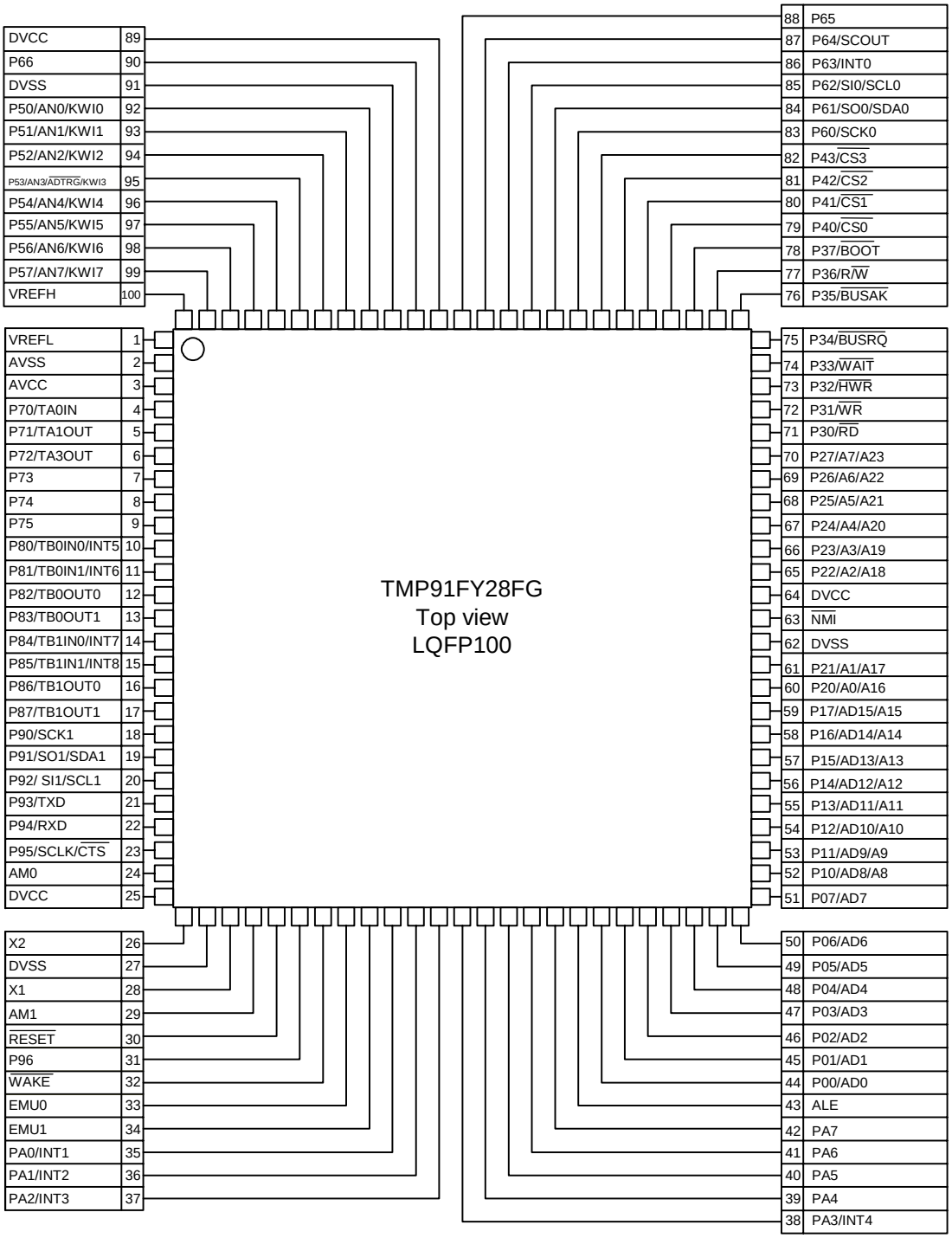


Figure 2.1.1 100-Pin LQFP Pin Assignment

2.2 Pin Usage Information

Table 2.2.1 to Table 2.2.4 list the input and output pins of the TMP91FY28, including alternate pin names and functions for multi-function pins.

Table 2.2.1 Pin names and functions (1/4)

Pin name	Number of Pins	I/O	Function
P00 to P07	8	I/O	Port 0: Individually programmable as input or output
AD0 to AD7		I/O	Address (Lower): Bits 0 to 7 of the address/data bus
P10 to P17	8	I/O	Port 1: Individually programmable as input or output
AD8 to AD15		I/O	Address/data (Upper): Bits 8 to 15 of the address/data bus
A8 to A15		Output	Address: Bits 8 to 15 of the address bus
P20 to P27	8	I/O	Port 2: Individually programmable as input or output
A0 to A7		Output	Address: Bits 0 to 7 of the address bus
A16 to A23		Output	Address: Bits 16 to 23 of the address bus
P30	1	Output	Port 30: Output only
$\overline{RD}$		Output	Read strobe: Asserted during a read operation from an external memory device Also asserted during a read from internal memory if P3<P30> = 0 and P3FC<P30F> = 1.
P31	1	Output	Port 31: Output only
$\overline{WR}$		Output	Write strobe: Asserted during a write operation on D0 to D7
P32	1	I/O	Port 32: Programmable as input or output (with internal pull-up resistor)
$\overline{HWR}$		Output	Higher write strobe: Asserted during a write operation on D8 to D15
P33	1	I/O	Port 33: Programmable as input or output (with internal pull-up resistor)
$\overline{WAIT}$		Input	Wait: Causes the CPU to suspend external bus activity ((1 + N) WAIT mode)
P34	1	I/O	Port 34: Programmable as input or output (with internal pull-up resistor)
$\overline{BUSRQ}$		Input	Bus request: Asserted by an external bus master to request bus mastership
P35	1	I/O	Port 35: Programmable as input or output (with internal pull-up resistor)
$\overline{BUSAK}$		Output	Bus acknowledge: Indicates that the CPU has relinquished the bus in response to $\overline{BUSRQ}$. (for external DMAC)
P36	1	I/O	Port 36: Programmable as input or output (with internal pull-up resistor)
R / $\overline{W}$		Output	Read/Write: Indicates the direction of data transfer on the bus: 1 = read or dummy cycle, 0 = write cycle
P37	1	I/O	Port 37: Programmable as input or output (with internal pull-up resistor)
$\overline{BOOT}$		Input	This pin is used to select single boot mode.
P40	1	I/O	Port 40: Programmable as input or output (with internal pull-up resistor)
$\overline{CS0}$		Output	Chip select 0: Asserted low to enable external devices at programmed addresses

Note: An external DMA controller configured with the BUSRQ and BUSAK pins cannot access the on-chip memory and peripheral functions of the TMP91FY28.

Table 2.2.2 Pin Names and Functions (2/4)

Pin name	Number of Pins	I/O	Function
P41 $\overline{CS1}$	1	I/O Output	Port 41: Programmable as input or output (with internal pull-up resistor) Chip select 1: Asserted low to enable external devices at programmed addresses
P42 $\overline{CS2}$	1	I/O Output	Port 42: Programmable as input or output (with internal pull-up resistor) Chip select 2: Asserted low to enable external devices at programmed addresses
P43 $\overline{CS3}$	1	I/O Output	Port 43: Programmable as input or output (with internal pull-up resistor) Chip select 3: Asserted low to enable external devices at programmed addresses
P50 to P57 AN0 to AN7 $\overline{ADTRG}$ KWI0 to KWI7	8	Input Input Input Input	Port 5: Input-only Analog input: Input to the on-chip AD converter AD trigger: Starts an AD conversion (Multiplexed with P53) Key wakeup input (Multiplexed with P50 to P57)
P60 SCK0	1	I/O I/O	Port 60: Programmable as input or output Clock input/output pin when the serial bus interface 0 is in SIO mode
P61 SO0 SDA0	1	I/O Output I/O	Port 61: Programmable as input or output (with internal pull-up resistor) Data transmit pin when the serial bus interface 0 is in SIO mode Data transmit/receive pin when the Serial Bus Interface 0 is in I^2C mode; programmable as an open-drain output
P62 SI0 SCL0	1	I/O Input I/O	Port 62: Programmable as input or output (with internal pull-up resistor) Data receive pin when the serial bus interface 0 is in SIO mode Clock input/output pin when the serial bus interface 0 is in I^2C mode; programmable as an open-drain output
P63 INT0	1	I/O Input	Port 63: Programmable as input or output Interrupt request 0: Programmable to be high-level, low-level, rising-edge or falling-edge sensitive
P64 SCOUT	1	I/O Output	Port 64: Programmable as input or output System clock output: Drives out f_{FPH} clock.
P65	1	I/O	Port 65: Programmable as input or output
P66	1	I/O	Port 66: Programmable as input or output
P70 TA0IN	1	I/O Input	Port 70: Programmable as input or output (with internal pull-up resistor) 8-bit timer 0 input: Input to timer 0
P71 TA1OUT	1	I/O Output	Port 71: Programmable as input or output (with internal pull-up resistor) 8-bit timer 1 output: Output from either timer 0 or timer 1
P72 TA3OUT	1	I/O Output	Port 72: Programmable as input or output (with internal pull-up resistor) 8-bit timer 3 output: Output from either timer 2 or timer 3

Table 2.2.3 Pin Names and Functions (3/4)

Pin name	Number of Pins	I/O	Function
P73	1	I/O	Port 73: Programmable as input or output (with internal pull-up resistor)
P74	1	I/O	Port 74: Programmable as input or output (with internal pull-up resistor)
P75	1	I/O	Port 75: Programmable as input or output (with internal pull-up resistor)
P80	1	I/O	Port 80: Programmable as input or output (with internal pull-up resistor)
TB0IN0		Input	16-bit timer 0 input 0: Count/capture trigger input to 16-bit timer 0
INT5		Input	Interrupt request 5: Programmable to be rising-edge or falling-edge sensitive
P81	1	I/O	Port 81: Programmable as input or output (with internal pull-up resistor)
TB0IN1		Input	16-bit timer 0 input 1: Capture trigger input to 16-bit timer 0
INT6		Input	Interrupt request 6: Rising-edge sensitive
P82	1	I/O	Port 82: Programmable as input or output (with internal pull-up resistor)
TB0OUT0		Output	16-bit timer 0 output 0: Output from 16-bit timer 0
P83	1	I/O	Port 83: Programmable as input or output (with internal pull-up resistor)
TB0OUT1		Output	16-bit timer 0 output 1: Output from 16-bit timer 0
P84	1	I/O	Port 84: Programmable as input or output (with internal pull-up resistor)
TB1IN0		Input	16-bit timer 1 Input 0: Count/capture trigger input to 16-bit timer 1
INT7		Input	Interrupt request 7: Programmable to be rising-edge or falling-edge sensitive
P85	1	I/O	Port 85: Programmable as input or output (with internal pull-up resistor)
TB1IN1		Input	16-bit timer 1 input 1: Capture trigger input to 16-bit timer 1
INT8		Input	Interrupt request 8: Rising-edge sensitive
P86	1	I/O	Port 86: Programmable as input or output (with internal pull-up resistor)
TB1OUT0		Output	16-bit timer 1 output 0: Output from 16-bit timer 1
P87	1	I/O	Port 87: Programmable as input or output (with internal pull-up resistor)
TB1OUT1		Output	16-bit timer 1 output 1: Output from 16-bit timer 1
P90	1	I/O	Port 90: Programmable as input or output
SCK1		I/O	Clock input/output pin when the serial bus interface 1 is in SIO mode
P91	1	I/O	Port 91: Programmable as input or output (with internal pull-up resistor)
SO1		Output	Data transmit pin when the serial bus interface 1 is in SIO mode
SDA1		I/O	Data transmit/receive pin when the serial bus interface 1 is in I^2C mode; programmable as an open-drain output
P92	1	I/O	Port 92: Programmable as input or output (with internal pull-up resistor)
SI1		Input	Data receive pin when the serial bus interface 1 is in SIO mode
SCL1		I/O	Clock input/output pin when the serial bus interface 1 is in I^2C mode; programmable as an open-drain output
P93	1	I/O	Port 93: Programmable as input or output
TXD		Output	Serial transmit data: Programmable as an open-drain output

Table 2.2.4 Pin Names and Functions (4/4)

Pin name	Number of Pins	I/O	Function
P94 RXD	1	I/O Input	Port 94: Programmable as input or output Serial receive data
P95 SCLK $\overline{\text{CTS}}$	1	I/O I/O Input	Port 95: Programmable as input or output Serial clock input/output Serial clear-to-send
P96	1	I/O	Port96: Programmable as input or output
PA0 to PA3 INT1 to INT4	4	I/O Input	Ports A0 to A3: Individually programmable as input or output (with internal pull-up resistor) Interrupt request 1 to 4: Individually programmable to be rising-edge or falling-edge sensitive
PA4 to PA7	4	I/O	Port A4 to A7: Programmable as input or output (with internal pull-up resistor)
$\overline{\text{WAKE}}$	1	Output	STOP mode monitor output This pin drives low when the CPU is operating; the pin is in high-impedance state during reset or in STOP mode.
ALE	1	Output	Address latch enable (This pin can be disabled in order to reduce noise.)
$\overline{\text{NMI}}$	1	Input	Non maskable interrupt request: Causes an NMI interrupt on the falling edge. Programmable to be rising-edge sensitive.
AM0 to AM1	2	Input	Both AM0 and AM1 should be held at logic 1.
EMU0	1	Output	Test pin. This pin should be left open.
EMU1	1	Output	Test pin. This pin should be left open.
$\overline{\text{RESET}}$	1	Input	Reset (with internal pull-up resistor): Initializes the whole TMP91FY28.
VREFH	1	Input	Input pin for high reference voltage for the AD converter.
VREFL	1	Input	Input pin for low reference voltage for the AD converter.
AVCC	1		Power supply pin for the AD converter.
AVSS	1		Ground pin for the AD converter.
X1/X2	2	I/O	Connection pins for an oscillator crystal
DVCC DVSS	3 3		Power supply pins. The DVCC pins should be connected to power supply. Ground pins. The DVSS pins should be connected to ground.

Note: All pins that have built-in pull-up resistors (Other than the $\overline{\text{RESET}}$ pin) can be disconnected from the built-in pull-up resistor by software.

3. Functional Description

This chapter describes the flash memory of the TMP91FY28, a flash version of the TMP91CW28. The TMP91FY28 contains a 256-Kbyte flash EEPROM and an 8-Kbyte RAM whereas the TMP91CW28 contains an 8-Kbyte RAM and 128-Kbyte ROM. In other respects, the hardware configuration and the functionality of the TMP91FY28 are identical to those of the TMP91CW28.

For descriptions of the configurations other than the flash memory, refer to the TMP91CW28 datasheet.

3.1 Overview of Operating Modes

The TMP91FY28 has the following two modes of operation.

The logic states on the $\overline{\text{BOOT}}$, AM0 and AM1 after a reset determine the mode of operation for the TMP91FY28.

Single Chip mode: The TMP91FY28 operates in Normal mode. After a reset, the TLCS-900/L1 core processor executes out of the on-chip flash memory.

Single Boot mode: After a reset, the TLCS-900/L1 core processor executes out of the on-chip boot ROM. The boot ROM contains a routine to aid users in performing on-board programming of the flash memory via a serial port (UART) of the TMP91FY28.

Table 3.1.1 Modes of Operation

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

3.2 Memory Map

Figure 3.2.1 shows memory assignment for the TMP91FY28 in single chip mode and the areas of memory the CPU can access in different addressing modes.

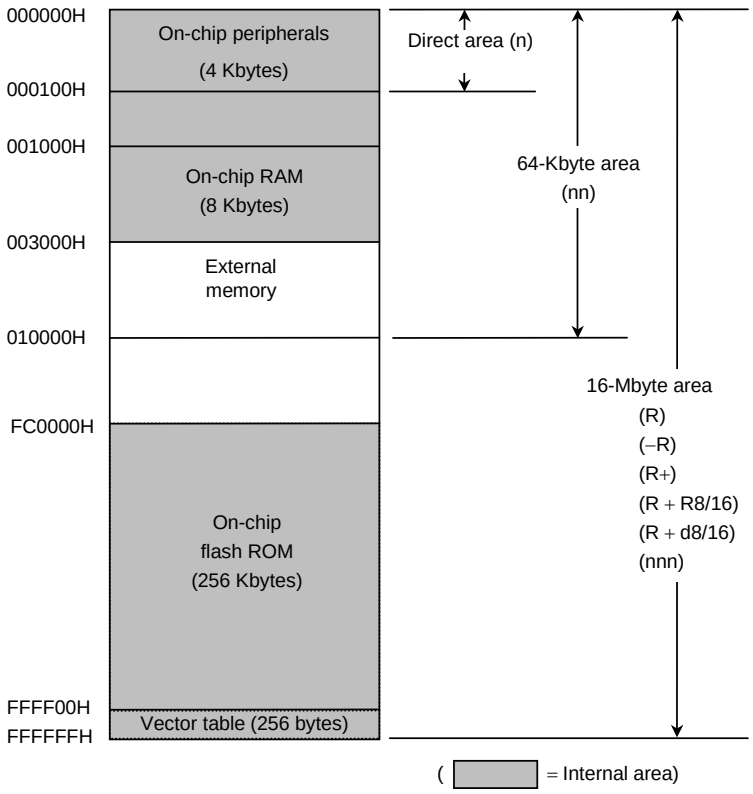


Figure 3.2.1 Memory Map for Single Chip Mode

3.3 Flash Memory

The TMP91FY28 contains flash memory that can be programmed or erased electrically using a single 2-V power supply.

Standard JEDEC commands are supported to program and erase the flash memory. Upon the entry of a command, the flash memory programs or erases its contents automatically. The flash memory can erase the entire chip at one time or erase the contents of one or more specified blocks.

Features

- Supply voltage in write/erase operation
  Vcc = 1.8 to 2.6 V
- Structure
  256 K × 8 bits/128 K × 16 bits (256 Kbytes)
- Functions
  Auto program
  Auto chip erase
  Auto block erase
  Auto multi-block erase
  DATA polling/Toggle bit
- Erase blocks
  16 Kbytes × 1/8 Kbytes × 2/
  32 Kbytes × 1/64 Kbytes × 3
- Command set compatible with the JEDEC EEPROM standard.
- General-purpose flash memory equivalent to the 29SL800TD
  * Some 29SL800TD functions, including block protection, are not supported.

Block architecture:

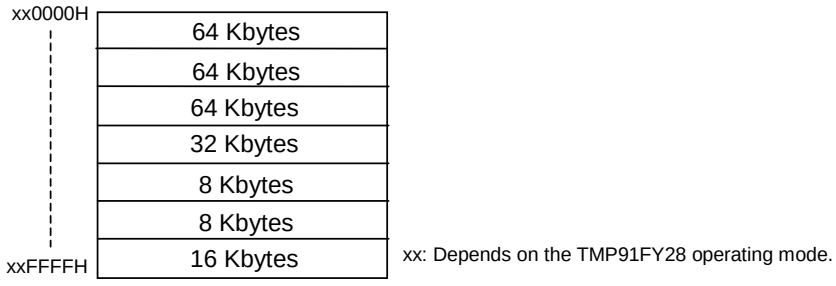


Figure 3.3.1 Block Architecture of Flash Memory

Command Definitions

Command Sequence	Cycles Required	Bus Cycles											
		1st Cycle (Write)		2nd Cycle (Write)		3rd Cycle (Write)		4th Cycle (Read/ Write)		5th Cycle (Write)		6th Cycle (Write)	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/Reset	1	XXXXXH	F0H										
Read/Reset	3	XAAAAH	AAH	x5554H	55H	xAAAAH	F0H	RA	RD				
Auto program	4	XAAAAH	AAH	x5554H	55H	xAAAAH	A0H	PA	PD				
Auto chip erase	6	XAAAAH	AAH	x5554H	55H	xAAAAH	80H	xAAAAH	AAH	x5554H	55H	xAAAAH	10H
Auto block erase	6	XAAAAH	AAH	x5554H	55H	xAAAAH	80H	xAAAAH	AAH	x5554H	55H	BA	30H

The addresses to be provided by the CPU are shown below.

Command Address	CPU Addresses: A23 to A0																
Address	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	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
X A A A A H	address area	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, written to DQ7 to DQ0

RA: Read address
RD: Read data
Data is read on a byte-by-byte or word-by-word basis.

PA: Program address
PD: Program data
The address must be even-numbered.
Data is written on a word-by-word basis.

BA: Block address. A combination of A17, A16, A15, A14, and A13 specifies an individual block.

*: Both types of reset commands can reset the device to read mode.

Write Status Flags

Status		DQ7	DQ6	DQ5	DQ3
Embedded operation in progress	Auto program	$\overline{DQ7}$	Toggle	0	0
	Auto erase (during the time-out window)	0	Toggle	0	0
	Auto erase	0	Toggle	0	1
Time-out in embedded operation	Auto program	$\overline{DQ7}$	Toggle	1	1
	Auto erase	0	Toggle	1	1

Note: DQ8 to DQ15, DQ0 to DQ2 are don't cares.

Block Erase Addresses

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

Basic Operations

The flash memory of the TMP91FY28 has the following two modes of operation:

- Read mode in which array data is read
- Embedded operation mode in which the flash memory is programmed or erased

The flash memory enters embedded operation mode when a valid command sequence is executed in read mode. In embedded operation mode, array data can not be read.

(1) Reading array data

The flash memory is automatically set to reading array data upon CPU reset after device power-up and after an embedded operation is successfully completed.

When an embedded operation is terminated abnormally, the read/reset command must be issued to put the flash memory back in Read mode as described below.

(2) Writing commands

The operations of the flash memory are selected by commands or command sequences written into the internal command register. This uses the same mechanism as for JEDEC-standard EEPROMs. Commands are made up of data sequences written at specific addresses via the command register.

The flash memory uses the command data provided via DQ0 to DQ7. It ignores any data appearing at DQ8 to DQ15.

The command sequence being written can be canceled by issuing the read/reset command between sequence cycles. The read/reset command clears the command register and resets the flash memory to read mode. Invalid command sequences also cause the flash memory to clear the command register and returns to read mode.

(3) Reset (Read/reset command)

The flash memory does not return to read mode if an embedded operation terminated abnormally. In this case, the read/reset command must be issued to put the flash memory back in read mode. The read/reset command may also be written between sequence cycles of the command being written to clear the command register.

(4) Auto program command

The flash memory is programmed on a word-by-word basis. As one word is 16 bits wide, the program address must be a multiple of two. The program address and data is latched in the fourth bus cycle of the auto program command sequence. The latching of the program data initiates the embedded auto program algorithm. The auto program command executes a sequence of internally timed events to program the desired bits of the addressed memory location and verify that the desired bits are sufficiently programmed. The system can determine the status of the programming operation by using write status flags.

Any commands written during the programming operation are ignored.

A bit must be programmed to change its state from a 1 to a 0. A bit cannot be programmed from a 0 back to a 1. Only an erase operation can change a 0 back to a 1.

If any failure occurs during the programming operation, the flash memory remains locked in embedded operation mode. The system can determine this status by using write status flags. To put the flash memory back in read mode, use the read/reset command to reset the flash memory. In case of a programming failure, it is recommended to discontinue the use of the failing flash block.

(5) Auto chip erase command

The embedded auto chip erase algorithm is initiated at the completion of the sixth bus cycle of a command sequence. The embedded auto chip erase algorithm automatically preprograms the entire memory for an all-0 data pattern prior to the erase; then, it automatically erases and verifies the entire memory for an all-1 data pattern. The system can determine the status of the chip erase operation by using write status flags.

Any commands written during the chip erase operation are ignored.

If any failure occurs during the erase operation, the flash memory remains locked in embedded operation mode. The system can determine this status by using write status flags. To put the flash memory back in read mode, use the read/reset command to reset the flash memory. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. The failing block can be identified by means of the auto block erase command.

(6) Auto block erase and auto multi-block erase commands

The address of the block to be erased is latched at the completion of the sixth bus cycle of a command sequence. After the time-out has expired, the erase operation will commence. The embedded auto block erase algorithm automatically preprograms the selected block for an all-0 data pattern, and then erases and verifies that block for an all-1 data pattern.

During the time-out period, additional block addresses and auto block erase commands may be written.

Any command other than auto block erase during the time-out period resets the flash memory to read mode. The block erase time-out period is 50 μm. The time-out window is reset at the completion of the sixth bus cycle. The system can determine the status of the erase operation by using write status flags.

Any commands written during the block erase operation are ignored.

If any failure occurs during the erase operation, the flash memory remains locked in embedded operation mode. The system can determine this status by using write status flags. To put the flash memory back in read mode, use the read/reset command to reset the flash memory. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block. If any failure occurred during the multi-block erase operation, the failing block can be identified by running auto block erase on each of the blocks selected for multi-block erasure.

(7) Write operation status

As shown in Table “Write status flags”, the flash memory provides several flag bits to determine the status of an embedded operation: DQ7, DQ6, DQ5 and DQ3. These status bits can be read during an embedded operation using the same timing as for read mode. The flash memory automatically returns to read mode when an embedded operation completes.

The system can determine the operating status by referencing write status flags during an embedded operation. Once an embedded operation has completed, the system can determine the status by checking whether the data it has read matches the cell data.

1. DQ7 (Data polling)

The data polling bit, DQ7, indicates to the host system the status of the embedded operation. Data polling is valid at the completion of the final bus cycle of a command sequence.

When the embedded program algorithm is in progress, an attempt to read the flash memory will produce the complement of the data last written to DQ7. Upon completion of the embedded program algorithm, an attempt to read the flash memory will produce the true data last written to DQ7. Therefore, the system can use DQ7 to determine whether the embedded program algorithm is in progress or completed.

When the embedded erase algorithm is in progress, an attempt to read the flash memory will produce a 0 at the DQ7 output. Upon completion of the embedded erase algorithm, the flash memory will produce a 1 at the DQ7 output.

If there is a failure during an embedded operation, DQ7 continues to drive out the same value.

The flash memory disables address latching when an embedded operation is complete. Data polling must be performed with a valid programmed address or an address within any of the non-protected blocks selected for erasure.

2. DQ6 (Toggle bit)

The toggle bit, DQ6, also indicates to the host system the status of the embedded operation.

Toggle bit is valid at the completion of the final bus cycle of a command sequence. Note that the erase operation will begin after the time-out has expired. When the embedded program algorithm is in progress, successive read cycles to any address cause DQ6 to toggle. If DQ6 is a 1 in the first read cycle, it will be a 0 in the next. Upon completion of the embedded program algorithm, DQ6 stops toggling and an attempt to read the flash memory will produce the data last written to DQ6. If there is a failure during an embedded operation, DQ6 still toggles.

3. DQ5 (Exceeded timing limits)

DQ5 produces a 0 while the program or erase operation is in progress normally. DQ5 produces a 1 to indicate that the program or erase time has exceeded the specified internal limit. This is a failure condition that indicates the program or erase cycle was not successfully completed.

The DQ5 failure condition also appears if the system tries to program a 1 to a location that was previously programmed to a 0. Only an erase operation can change a 0 back to a 1. In this case, the embedded program algorithm halts the operation. Once the operation has exceeded the timing limits, DQ5 will indicate a 1. Note that this is not a device failure condition since the flash memory was used incorrectly.

Under both these conditions, the flash memory remains locked in embedded operation mode. The system must issue the read/reset command to return the flash memory to read mode.

4. DQ3 (Block erase timer)

The block erase time-out window begins at the completion of the sixth bus cycle of the command sequence. The erase operation will begin after the time-out has expired (80 μs). When the time-out is complete and the erase operation has begun, DQ3 switches from 0 to 1. If DQ3 is 0, the flash memory will accept additional auto block erase commands. Each time an auto block erase command is written, the time-out window is reset. To ensure that the command has been accepted, the system should check DQ3 prior to and following each auto block erase command. If DQ3 is 1 on the second status check, the command might not have been accepted.

5. RDY/$\overline{\text{BSY}}$ (Ready/busy)

This signal is not available because it is not connected to the CPU.

(8) Re-programming the flash memory from the internal CPU

The internal CPU can re-program the flash memory using the command sequence and write status flags described above. Because the flash memory cannot be read while it is performing an embedded operation, the programming routine must be executed in a memory area other than those assigned to the flash memory.

The internal CPU can re-program the flash memory in one of two modes: using single boot mode or using a user-defined protocol in single chip mode (User boot mode).

1. Single boot mode

In single boot mode, the flash memory can be re-programmed by using a program contained in the TMP91FY28 on-chip boot ROM. This boot ROM is a masked ROM. When single boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it (See Figure 3.4.2 on page 23).

The program in the boot ROM fetches new application data by serial transfer and re-programs the flash memory with that data. Interrupts should be disabled in single boot mode, including the $\overline{\text{NMI}}$ and other nonmaskable interrupts.

For details, see section 3.4, “Single Boot Mode”.

2. User boot mode (Single chip mode)

User boot mode allows you to create a programming algorithm of your own. User boot mode is a sub-mode of single chip mode, or normal mode. This mode also requires that the flash programming routine run in address space outside the flash memory area and that all interrupts, including nonmaskable interrupts, be disabled.

The user must provide a flash programming routine, including a routine for fetching new application data, with which the flash will be re-programmed. Code the main program so that it can switch from normal operation to flash memory programming mode, in which it expands and executes the flash programming routine outside the flash memory area. A flash programming routine may be stored in the flash memory and expanded into the on-chip RAM for execution or it may be stored and executed in an external memory device.

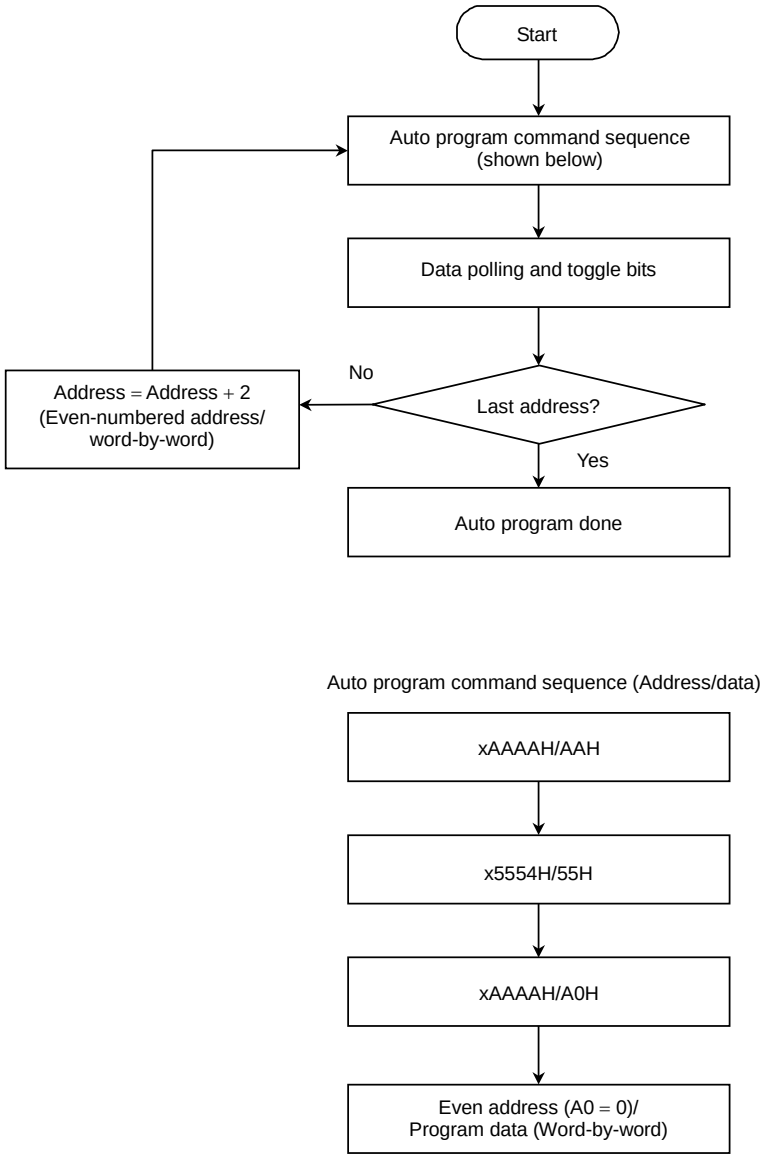


Figure 3.3.2 Auto Security on Operation

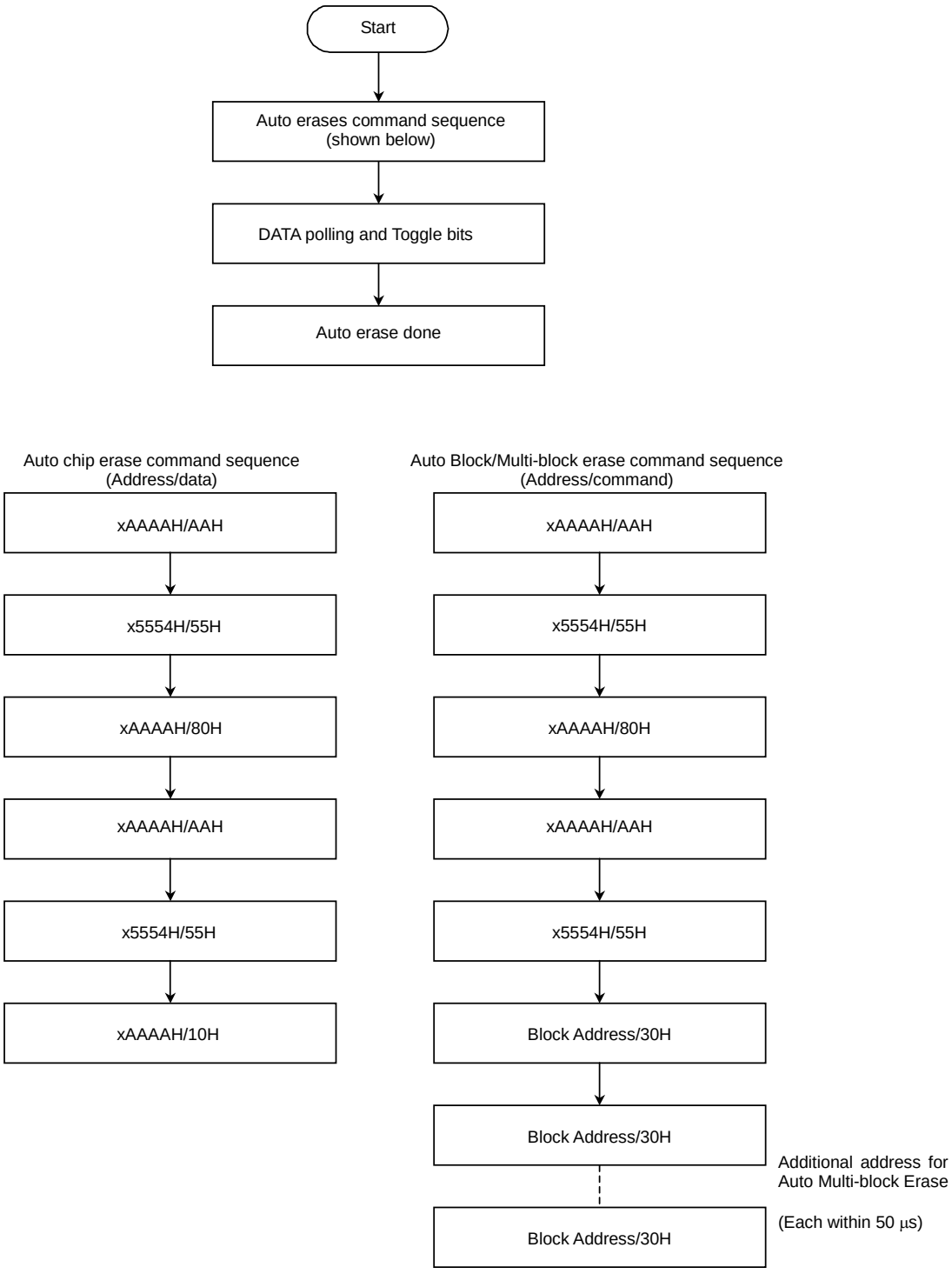


Figure 3.3.3 Auto Erase Operations

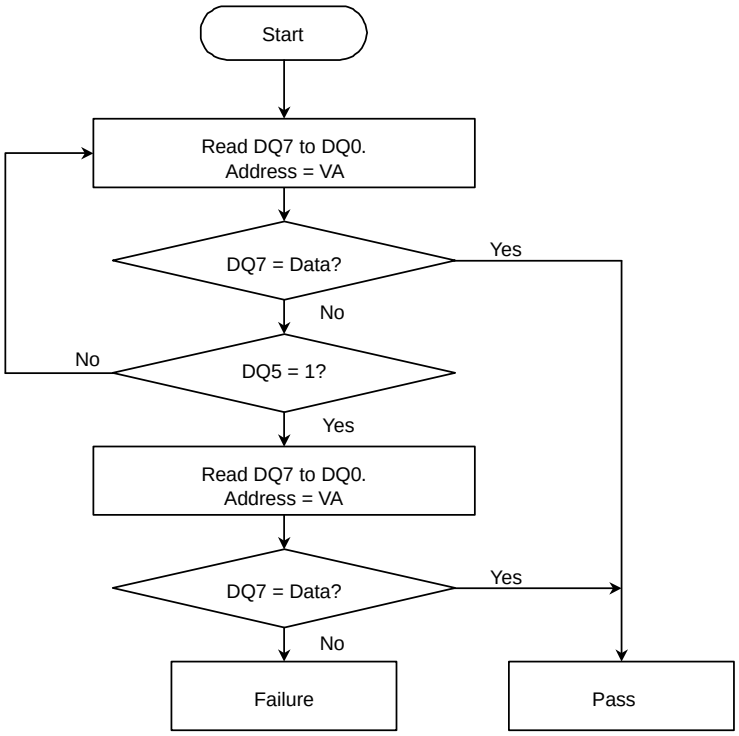


Figure 3.3.4 Data Polling (DQ7) Algorithm

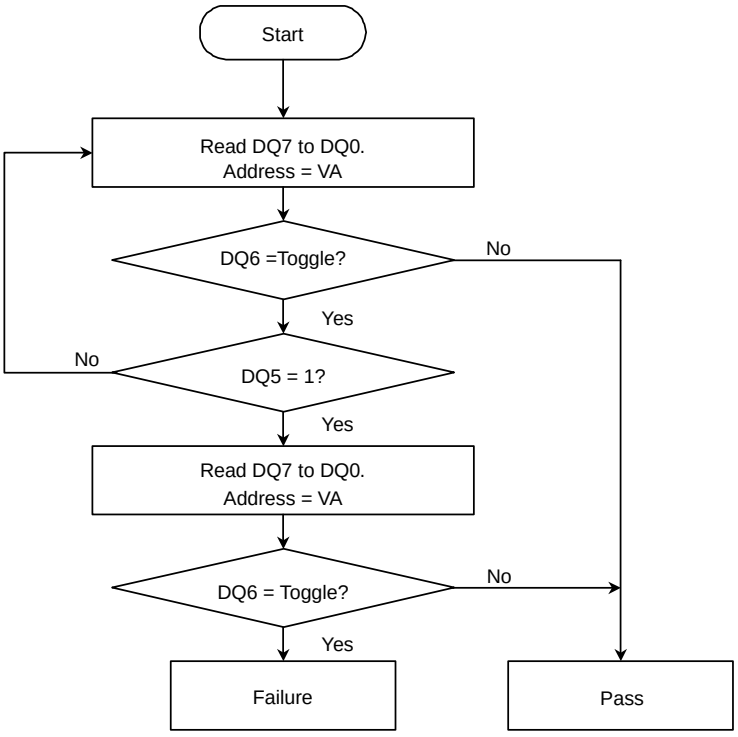


Figure 3.3.5 Toggle Bit Algorithm (DQ6)

VA: Auto program: The address at which data is being written
Auto chip erase: Any flash memory address
Auto block erase: The selected block address

3.4 Single Boot Mode

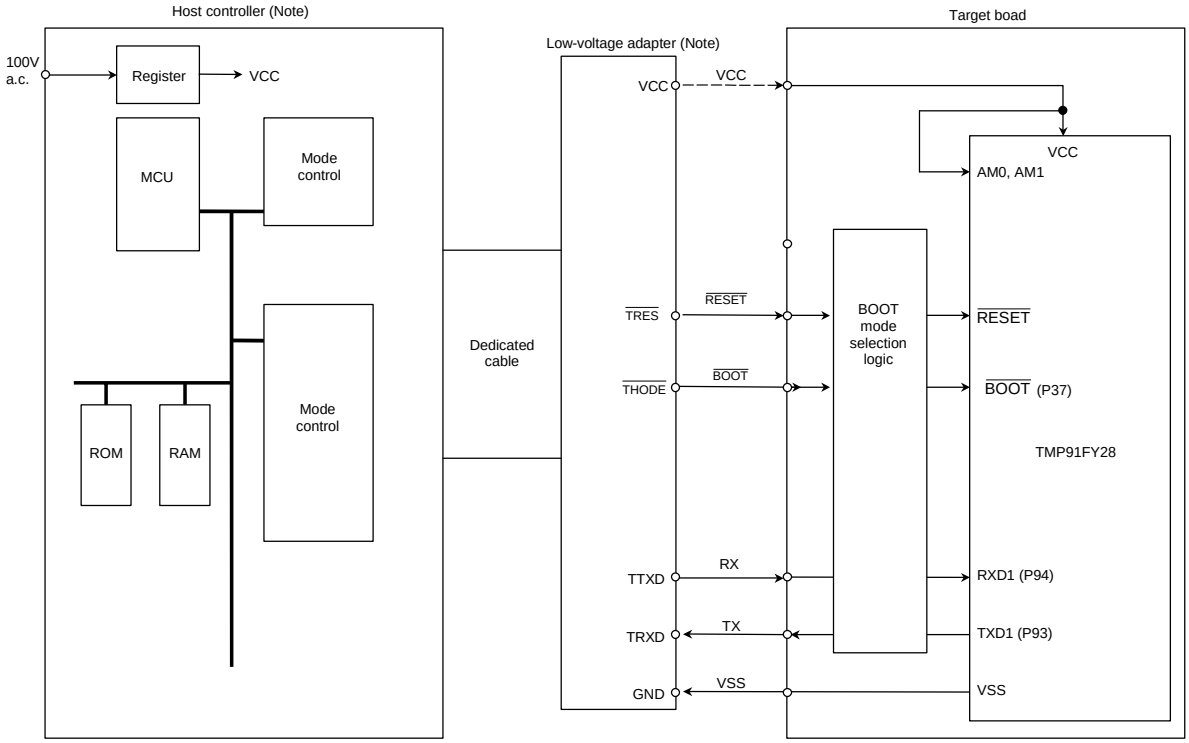
(1) Overview

The TMP91FY28 has single boot mode for serial programming of the flash memory while the TMP91FY28 is installed on the board. When single boot mode is selected, the boot ROM is mapped to an address region. In single boot mode, the flash memory can be re-programmed by using a program contained in the on-chip boot ROM. This boot ROM is a masked ROM.

For on-board programming, the SIO of the TMP91FY28 is connected to an external host controller, which issue commands to the target board.

The boot program contained in the boot ROM offers RAM transfer command, which stores program code transferred from a host controller to the on-chip RAM.

Figure 3.4.1 shows an example of host-to-target connection.



Note: The AF210 (Advanced on-board flash microcomputer programmer) and the AF264 (Voltage conversion adapter) from Yokogawa Digital Computer Corporation are supported. For a detailed description, consult the manual that accompanies the AF210 and AF264.
Contact: Yokogawa Digital Computer Corporation
Instrument Business Division
Phone: +81-42-333-6224

Figure 3.4.1 Example of a Connection Between a Host Controller and a Target Board

(2) Configuring for single boot mode

For on-board programming, boot the TMP91FY28 in single boot mode, as follows:

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

Set the AM0, AM1 and $\overline{\text{BOOT}}$ inputs as the logic values shown above. The TMP91FY28 boots in single boot mode on the rising edge of the $\overline{\text{RESET}}$ pin.

(3) Memory map

Figure 3.4.2 shows a comparison of the memory maps in user boot and single boot modes. In single boot mode, the on-chip flash memory is mapped to physical addresses 10000H through 4FFFFH, and the on-chip boot ROM (Masked ROM) is mapped to physical addresses FFF800H through FFFFFFH.

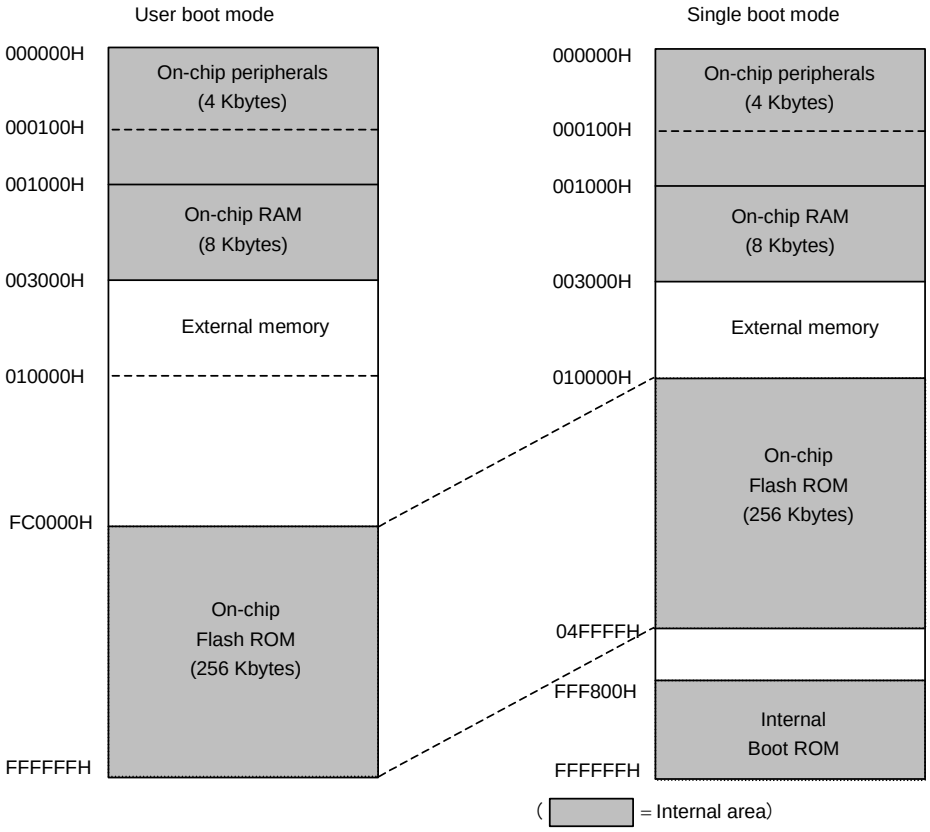


Figure 3.4.2 Memory Maps for User Boot and Single Boot Modes

(4) Interface specification

The following shows specifications for SIO communication in single boot mode.

To enable on-board programming, the host controller must be configured to use these specifications.

The baud rate is initially 9600 bps, which can be changed as shown in Table 3.4.1.

Cominnucation channel:	SIO channel 1
Serial transfer mode:	UART (Asynchronous) mode, full-duplex
Data length:	8 bits
Pality bit:	None
STOP bit:	1
Baud rate (reset value)	9600 bps

(5) Data transfer format

Table 3.4.1 to Table 3.4.7 show baud rate change codes, operation commands, and data transfer formats in different operating modes. Also refer to "Description of the boot program commands," following the tables.

Table 3.4.1 Baud Rate Change Codes

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

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

Table 3.4.2 Single Boot Mode Commands

Code	Command
30H	Program flash
60H	RAM transfer
90H	Show flash memory SUM

Table 3.4.3 Operating Frequency and Baud Rate in Single Boot Mode

Reference Baud Rate (bps)		9600		19200		31250		38400		57600		62500		76800	
Baud Rate Change Code		28H		18H		0AH		07H		06H		05H		04H	
Reference Frequency (MHz)	Area (MHz)	Baud rate (bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)
4.9152	4.85 to 5.07	9600	0	19200	0	—	—	38400	0	—	—	—	—	76800	0
5		9766	+1.73	19531	+1.72	—	—	39063	+1.73	—	—	—	—	78125	+1.73
6	5.91 to 6.23	9375	−2.34	18750	−2.34	31250	0	—	—	—	—	—	—	—	—
6.144		9600	0	19200	0	32000	+2.4	—	—	—	—	—	—	—	—
7.3728	7.26 to 7.48	9600	0	19200	0	—	—	38400	0	57600	0	—	—	—	—
8	7.84 to 8.16	9615	+0.16	—	—	31250	0	—	—	—	—	62500	0	—	—
9.8304	9.64 to 10.20	9600	0	19200	0	30720	−1.7	38400	0	—	—	—	—	76800	0
10		9766	+1.73	19531	+1.72	31250	0	39063	+1.73	—	—	—	—	78125	+1.73
12	11.76 to 12.75	9375	−2.34	18750	−2.34	31250	0	37500	−2.34	—	—	62500	0	—	—
12.288		9600	0	19200	0	32000	+2.4	38400	0	—	—	64000	+2.4	—	—
12.5		9766	+1.73	19531	+1.72	32552	+4.17	39063	+1.73	—	—	65104	+4.17	—	—
14.7456	14.46 to 15.04	9600	0	19200	0	32914	+5.3	38400	0	57600	0	—	—	76800	0
16	15.68 to 16.32	9615	+0.16	19231	+0.16	31250	0	—	—	—	—	62500	0	—	—
18	17.64 to 18.36	9375	−2.34	18750	−2.34	31250	0	—	—	56250	−2.34	—	—	—	—
19.6608	19.27 to 20.40	9600	0	19200	0	30720	−1.7	38400	0	—	—	61440	−1.7	76800	0
20		9766	+1.73	19531	+1.72	31250	0	39063	+1.73	—	—	62500	0	78125	+1.73
21.18	20.76 to 22.56	9193	−4.24	18385	−4.24	30085	−3.73	36771	−4.24	55156	−4.24	—	—	—	—
22.1184		9600	0	19200	0	31418	+0.54	38400	0	57600	0	—	—	—	—
24.5760	24.09 to 25.50	9600	0	19200	0	32000	+2.4	38400	0	54857	−4.76	64000	+2.4	76800	0
25		9766	+1.73	19531	+1.72	32552	+4.17	39063	+1.73	55804	−3.12	65104	+4.17	78125	+1.73
26.88	26.35 to 27.54	9545	−0.57	19091	−0.57	30000	−4	38182	−0.57	—	—	—	—	—	—
27		9588	−0.13	19176	−0.13	30134	−3.57	38352	−0.13	—	—	—	—	—	—

Reference frequency: High-speed oscillator frequencies supported in single boot mode
When re-programming the flash memory in single boot mode, select any of the reference frequencies for the high-speed clock.

Area: Approximate range of clock frequencies that are detected as each reference frequency. Single boot operation may be disabled at clock frequencies not included in any of the detectable ranges.

Note: To automatically detect a reference frequency (Microcontroller clock frequency), the total error must be within ±3%, including the transmission baud rate error (at 9600 bps) for the host controller, the oscillation frequency error, and the matching data timing detection error.

Table 3.4.4 Format of Data Transfer by the Boot Program (for re-programming the flash memory)

	Byte	Data Transferred from the Controller to TMP91FY28	Baud Rate	Data Transferred from the TMP91FY28 to Controller
Boot ROM	1st byte 2nd byte	Matching data (5AH)	9600 bps 9600 bps	(Baud rate set automatically) OK: Echo back data (5AH) Error: None
	3rd byte 4th byte	Baud rate change code (See Table 3.4.1) –	9600 bps 9600 bps	– OK: Echo back data Error: A1H × 3, A2H × 3, A3H × 3, 62H × 3 (*1)
	5th byte 6th byte	Command code (30H) –	New baud rate New baud rate	– OK: Echo back data (30H) Error: A1H × 3, A2H × 3, A3H × 3, 63H × 3 (*1)
	7th byte	–	New baud rate	OK: C1H Error: 64H × 3 (*1)
	8th byte : (n-2)th byte	Data in Intel hexadecimal object file format (Binary) (*2)	New baud rate	–
	(n-1)th byte	–	New baud rate	OK: SUM (Upper byte) (*3) Error: None
	nth byte	–	New baud rate	OK: SUM (Lower byte) (*3) Error: None
	(n+1)th byte	(Wait for the next command code.)	New baud rate	–

*1: "xxH × 3" means that the boot program transmits three bytes of xxH and then stops operating. See "Code transmitted by the boot program," described later in this section.

*2: See "Notes on Intel hexadecimal object file format (Binary)," described later in this section.

*3: See "Calculation of the Show Flash Memory SUM Command," described later in this section.

Table 3.4.5 Format of Data Transfer by the Boot Program (for RAM transfer)

	Byte	Data Transferred from the Controller to TMP91FY28	Baud Rate	Data Transferred from the TMP91FY28 to Controller
Boot ROM	1st byte 2nd byte	Matching data (5AH) –	9600 bps 9600 bps	– (Baud rate set automatically) OK: Echo back data (5AH) Error: None
	3rd byte 4th byte	Baud rate change code (See Table 3.4.1) –	9600 bps 9600 bps	– OK: Echo back data Error: A1H × 3, A2H × 3, A3H × 3, 62H × 3 (*1)
	5th byte 6th byte	Command code (60H) –	New baud rate New baud rate	– OK: Echo back data (60H) Error: A1H × 3, A2H × 3, A3H × 3, 63H × 3 (*1)
	7th byte 8th byte	Password count storage address bits 23 to 16 (*2) –	New baud rate New baud rate	– OK: None Error: A1H ×3, A2H × 3, A3H × 3 (*1)
	9th byte 10th byte	Password count storage address bits 15 to 08 (*2) –	New baud rate New baud rate	– OK: None Error: A1H ×3, A2H × 3, A3H × 3 (*1)
	11th byte 12th byte	Password count storage address bits 07 to 00 (*2) –	New baud rate New baud rate	– OK: None Error: A1H ×3, A2H × 3, A3H × 3 (*1)
	13th byte 14th byte	Password comparison start address bits 23 to 16 (*2) –	New baud rate New baud rate	– OK: None Error: A1H ×3, A2H × 3, A3H × 3 (*1)
	15th byte 16th byte	Password comparison start address bits 15 to 08 (*2) –	New baud rate New baud rate	– OK: None Error: A1H ×3, A2H × 3, A3H × 3 (*1)
	17th byte 18th byte	Password comparison start address bits 07 to 00 (*2) –	New baud rate New baud rate	– OK: None Error: A1H ×3, A2H × 3, A3H × 3 (*1)
	19th byte : mth byte	Password sequence (*2) –	New baud rate New baud rate	– OK: None Error: A1H ×3, A2H × 3, A3H × 3 (*1)
	(m+1)th byte : (n-2)th byte	Data in Intel hexadecimal object file format (Binary) (*3)		–
	(n-1)th byte	–	New baud rate	OK: SUM (Upper byte) (*4) Error: None
	nth byte	–	New baud rate	OK: SUM (Lower byte) (*4) Error: None
RAM	–	Branch to the user program start address.		

*1: "xxH × 3" means that the boot program transmits three bytes of xxH and then stops operating. See "Code transmitted by the boot program," described later in this section.

*2: Refer to "Notes on passwords," described later in this section.

*3: See "Notes on Intel hexadecimal object file format (Binary)," described later in this section.

*4: See "Calculation of the Show Flash Memory SUM Command," described later in this section.

Table 3.4.6 Format of Data Transfer by the Boot Program (for the flash memory SUM)

	Byte	Data Transferred from the Controller to TMP91FY28	Baud Rate	Data Transferred from the TMP91FY28 to Controller
Boot ROM	1st byte 2nd byte	Matching data (5AH) –	9600 bps 9600 bps	– (Baud rate set automatically) OK: Echo back data (5AH) Error: None
	3rd byte 4th byte	Baud rate change code (See Table 3.4.1) –	9600 bps 9600 bps	– OK: Echo back data Error: A1H × 3, A2H × 3, A3H × 3, 62H × 3 (*1)
	5th byte 6th byte	Command code (90H) –	New baud rate New baud rate	– OK: Echo back data (90H) Error: A1H × 3, A2H × 3, A3H × 3, 63H × 3 (*1)
	7th byte	–	New baud rate	OK: SUM (Upper byte) (*2) Error: –
	8th byte	–	New baud rate	OK: SUM (Lower byte) (*2) Error: –
	9th byte	(Wait for the next command code.)	New baud rate	–

*1: "xxH × 3" means that the boot program transmits three bytes of xxH and then stops operating. See "Code transmitted by the boot program," described later in this section.

*2: See "Calculation of the Show Flash Memory SUM Command," described later in this section.

(6) Description of the boot program commands

When the TMP91FY28 is started in single boot mode, the boot program runs automatically. The boot program offers the following three commands, the details of which are provided on the following subsections.

1. Program flash command

The program flash command first erases the entire flash memory chip (256 Kbytes) and then writes data to specified addresses. The host controller must transmit write data as binary data in Intel hexadecimal object file format.

Once all records have been written without an error, the boot program calculates the SUM of 256 Kbytes in the flash memory and returns the result.

2. RAM transfer command

The RAM transfer command stores Intel hexadecimal object file format data transferred from the host controller to the on-chip RAM. Once the transfer is successfully completed, the boot program calculates and transmits the SUM, and then starts executing the user program. The address received first specifies the address at which the user program should start.

The RAM transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner.

The programming routine must utilize the flash memory command sequences described earlier in this section (to align with flash memory addresses used in single boot mode).

Before initiating a transfer, the RAM transfer command checks a password sequence coming from the controller against that stored in the flash memory. If they do not match, the RAM transfer command aborts.

3. Show flash memory SUM command (See Table 3.4.4)

The show flash memory SUM command adds the contents of the 256 Kbytes of the flash memory together. The boot program does not provide a command to read out the contents of the flash memory. Instead, the flash memory SUM command can be used for software revision management.

a. Program flash command

1. The 1st byte specifies matching data. Once started in single boot mode, the boot program waits for matching data to be transmitted from the host controller. Upon the reception of matching data, the program automatically adjusts the initial baud rate for the serial channel to 9600 bps. Matching data is 5AH.
2. The 2nd byte, transmitted from the TMP91FY28 to the controller, is an acknowledge response to the 1st byte. After setting the baud rate automatically, the boot program echoes back the 1st byte (5AH). If it fails to set the baud rate, it stops operation.

3. The 3rd byte specifies a new baud rate, which is one of the seven supported baud rates listed in Table 3.4.1. If the controller need not change the baud rate, it must transmit the default baud rate (28H: 9600 bps). The new baud rate does not become effective until the boot program echoes back the data.
4. The 4th byte, transmitted from the TMP91FY28 to the controller, is an acknowledge response to the 3rd byte. If the received data corresponds to any of the baud rates supported for the current operating frequency, the boot program echoes back the data before changing the baud rate. Otherwise, it transmits three bytes of error code (62H) and then stops operation.
5. The 5th byte, which the TMP91FY28 receives from the controller, is a command. The code for the program flash command is 30H.
6. The 6th byte, transmitted from the TMP91FY28 to the controller, is an acknowledge response to the 5th byte. If the 5th byte is equal to any of the command codes listed in Table 3.4.2, the boot program echoes it back to the controller. When the program flash command was received, the boot program echoes back a value of 30H and then branches to the flash programming routine. If the 5th byte is not a valid command, the boot program sends back three bytes of error code (63H) and then stops operation.
7. The 7th byte, transmitted from the TMP91FY28 to the controller, indicates whether chip erase operation (256 Kbytes) has completed successfully. If the chip has been erased normally, the flash programming routine transmits the normal erase completion code (C1H).
   If an error occurs during erasure, the routine transmits three bytes of error code (64H) and then stops operation.
   The controller can transmit next data once it receives the normal erase completion code (C1H).
8. The 8th to (n–2)th bytes, which the TMP91FY28 receives from the controller, are binary data in Intel hexadecimal object file format. The TMP91FY28 does not echo back these bytes.
   The flash programming routine ignores received data, without transmitting an error code, until it detects a RECORD MARK for Intel hexadecimal object file format (3AH, ":"). Once it detects a RECORD MARK, it receives a sequence from the RECLEN field to CHKSUM field. The routine sequentially writes each received byte to a specified flash memory address. The first record must be an extended segment address record because bits 23 to 16 of the write address pointer are 00H by default.
   Once the routine has received a single record, from the RECORD MARK to CHKSUM field, it again waits for a RECORD MARK.
   If a write error, reception error, or Intel hexadecimal object file format error occurs, the routine stops operation without transmitting an error code.
   The flash programming routine executes the show flash memory SUM routine when it detects an end of file record. The controller must, therefore, wait for the SUM after transmitting an end of file record.

9. The show flash memory SUM routine adds all the bytes of the flash memory together. The (n–1)th and nth bytes, transmitted from the TMP91FY28 to the controller, indicate the upper and lower bytes of the total SUM, respectively. For details, see section, "Calculation of the Show Flash Memory SUM Command." The SUM is calculated only when the end of file record is detected without a write error, reception error, or Intel hexadecimal object file format error. Calculating the SUM for the 256-Kbyte flash memory area requires approximately 400 ms at fc = 20 MHz. After calculating the SUM, the program transmits the SUM to the controller. After transmitting the end of file record, the controller can determine whether the re-programming of the flash memory has completed successfully, according to whether it receives the SUM.
10. The (n+1)th byte will be the next command code if re-programming completes successfully.

b. RAM transfer command (See Table 3.4.5.)

1. The processing of the 1st to 4th bytes are the same as for the program flash command.
2. The 5th byte, which the TMP91FY28 receives from the controller, is a command. The code for the RAM transfer command is 60H.
3. The 6th byte, transmitted from the TMP91FY28 to the controller, is an acknowledge response to the 5th byte. If the 5th byte is equal to any of the command codes listed in Table 3.4.2, the boot program echoes it back to the controller. When the RAM transfer command was received, the boot program echoes back a value of 60H and then branches to the RAM transfer routine. If the 5th byte is not a valid command, the boot program sends back three bytes of error code (63H) and then stops operation.
4. The 7th byte contains data for bits 23 to 16 of the address storing the number of passwords. The address is specified using three bytes. Note that operation is canceled if the received passwords are less than eight.
5. The 8th byte, from the TMP91FY28 to the controller, is not transmitted if the 7th byte has been received without an error. If a reception error occurs, the RAM transfer routine transmits three bytes of error code and then stops operation.
6. The 9th to 12th bytes correspond to data for bits 15 to 8 and 7 to 0 of the password count storage address and the respective error code bytes, if any. See steps 4 and 5, above.
7. The 13th byte contains data for bits 23 to 16 of the address at which the comparison of passwords will start. The address is specified using three bytes.
8. The 14th byte, from the TMP91FY28 to the controller, is not transmitted if the 13th byte has been received without an error. If a reception error occurs, the RAM transfer routine transmits three bytes of error code and then stops operation.
9. The 15th to 18th bytes correspond to data for bits 15 to 8 and 7 to 0 of the password comparison start address and the respective error code bytes, if any. See steps 7 and 8, above.
10. The 19th to mth bytes contain passwords. The number of passwords (N) is specified with the data stored at the password count storage address. The RAM transfer routine compares N passwords with those stored in the area starting with the password comparison start address. The controller must transmit N bytes of password data. If any of the passwords fails to match in comparison, the routine stops operation without transmitting an error code.

11. The (m+1)th to (n–2)th bytes, which the TMP91FY28 receives from the controller, are binary data in Intel hexadecimal object file format. The TMP91FY28 does not echo back these bytes.
    The RAM transfer routine ignores received data, without transmitting an error code, until it detects a RECORD MARK for Intel hexadecimal object file format (3AH, ":"). Once it detects a RECORD MARK, it receives a sequence from the RECLEN field to CHKSUM field. The routine sequentially writes each received byte to a specified RAM address. Bits 23 to 16 of the write address pointer are 00H by default. The first record need not be an extended segment address record. Once the routine has received a single record, from the RECORD MARK to CHKSUM field, it again waits for a RECORD MARK.
    If a reception error or Intel hexadecimal object file format error occurs, the routine stops operation without transmitting an error code.
    The RAM transfer routine executes the show flash memory SUM routine when it detects an end of file record. The controller must, therefore, wait for the SUM after transmitting an end of file record.
12. The (n–1)th and nth bytes, transmitted from the TMP91FY28 to the controller, indicate the upper and lower bytes of the SUM, respectively. For details, see section, "Calculation of the Show Flash Memory SUM Command." The SUM is calculated only when the end of file record is detected without a reception error or Intel hexadecimal object file format error. The time required for calculating the SUM is roughly proportional to the number of data bytes written to RAM. Calculating the SUM for a 4-Kbyte RAM area requires approximately 6 ms at fc = 20 MHz. After calculating the SUM, the program transmits the SUM to the controller. After transmitting the end of file record, the controller can determine whether transfer to the RAM has completed successfully, according to whether it receives the SUM.
13. After transmitting the SUM, the program makes a branch to the address specified with the first data byte received in Intel hexadecimal object file format.

c. Show flash memory SUM command (See Table 3.4.6.)

1. The processing of the 1st and 4th bytes are the same as for the program flash command.
2. The 5th byte, which the target board receives from the controller, is a command. The code for the show flash memory SUM command is 90H.
3. If the 6th byte is equal to any of the command codes listed in Table 3.4.2 on page 24, the boot program echoes it back to the controller. When the show flash memory SUM command was received, the boot program echoes back a value of 90H and then branches to the show flash memory SUM routine.
   If the 6th byte is not a valid command, the boot program sends back 63H to the controller and then stops operation.
4. The show flash memory SUM routine adds all the bytes of the flash memory together. The 7th and 8th bytes, transmitted from the target board to the controller, indicate the upper and lower bytes of this total SUM, respectively. For details, see section, "Calculation of the Show Flash Memory SUM Command".
5. The 9th byte is the next command code.

d. Code transmitted by the boot program

The boot program represents processing states with specific codes, as listed below.

Table 3.4.7 Code Transmitted by the Boot Program

Code	Description
C1H	Chip erase operation has completed successfully.
62H, 62H, 62H	A baud rate change error has occurred.
63H, 63H, 63H	A command error has occurred.
64H, 64H, 64H	An erase error has occurred.
A1H, A1H, A1H	Received data contains a framing error. (Note)
A2H, A2H, A2H	Received data contains a parity error. (Note)
A3H, A3H, A3H	Received data contains an overrun error. (Note)

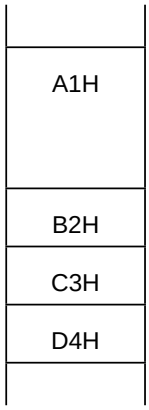
Note: If any of these errors occur while data in Intel hexadecimal object file format is being received, the boot program does not transmit a reception error code.

e. Calculation of the show flash memory SUM command

1. Calculation method

The show flash memory SUM command adds all 256 Kbytes of the flash memory together and provides the total SUM as a word quantity.

Example:



For the interest of simplicity, assume the depth of the flash memory is four locations. Then the SUM of the four bytes is calculated as:

A1H + B2H + C3H + D4H = 02EAH

SUM (High) = 02H
SUM (Low) = EAH

When the program flash, RAM transfer, and show flash memory SUM commands are executed, the SUM is calculated as described above.

2. Scope of calculation

Table 3.4.8 lists the data to be totaled to obtain the SUM.

Table 3.4.8 Scope of SUM Calculation

Command	Data to be Totaled	Remarks
Program flash command	Data stored in the entire flash memory area (256 Kbytes)	The data to be totaled is not limited to the data actually written to the flash memory or RAM. If received addresses are not contiguous, leaving some intermediate areas unwritten, those areas are also included in the calculation of the SUM.
RAM transfer command	Data written to the area from the address received first to that received last	
Show flash memory SUM command	Data stored in the entire flash memory area (256 Kbytes)	–

f. Notes on Intel hexadecimal object file format (Binary)

1. The program flash command requires that the first record be an extended segment address record. This is because the TMP91FY28 flash memory is mapped to an area starting from address 10000H and bits 23 to 16 of the write address pointer are 00H by default.
2. The RAM transfer command does not require that the first record be an extended segment address record. This is because bits 23 to 16 of the write address pointer are 00H by default.
3. After receiving the CHKSUM field of a record, the program waits for a RECORD MARK (3AH, ":") for the next record. If any data other than 3AH is transmitted between records, it is ignored.

Note: ":": 3AH (RECORD MARK)
xx, yy: Data written to flash memory
CS, EC, DC, FF: Checksum data
zz: No effect if transmitted by the controller
ww: Must not be transmitted by the controller

4. After transmitting the CHKSUM field of an end of file record, the program on the controller must wait for two bytes of data (Upper and lower bytes of the SUM) to be received, without transmitting any data. After receiving the CHKSUM field of an end of file record, the SUM calculation routine on the TMP91FY28 calculates the SUM and transmits the result as two bytes.
5. If a write error (only for the program flash command), reception error, or Intel hexadecimal object file format error occurs, the program stops operation without transmitting an error code. An Intel hexadecimal object file format error occurs in the following cases:

- The RECTYP field of a record is other than 00H, 01H, and 02H.
- A checksum error occurs.
- The RECLEN field of an extended segment address record (RECTYP = 02H) is other than 02H.
- The LOAD OFFSET field of an extended segment address record (RECTYP = 02H) is other than 0000H.
- The second byte of the data contained in an extended segment address record (RECTYP = 02H) is other than 00H.
- The RECLEN field of an end of file record (RECTYP = 01H) is other than 00H.
- The LOAD OFFSET field of an end of file record (RECTYP = 01H) is other than 0000H.

Example: Table 3.4.9 shows the transfer format when writing data to memory space in the address range of 1FFF8H to 2002FH.

Table 3.4.9 Example Transfer Format for the Program Flash Command

Direction of Transfer	Meaning of Data Intel Hexadecimal Object File Format (8th to (n−2)th bytes in Table 3.4.4)	Data
Controller to TMP91FY28	Extended segment address record	: 02 0000 02 1000 EC zz
Controller to TMP91FY28	Data record (Data length: 08H)	: 08 FFF8 00 xxxxxx CS zz
Controller to TMP91FY28	Extended segment address record	: 02 0000 02 2000 DC zz
Controller to TMP91FY28	Data record (Data length: 30H)	: 30 0000 00 yyyyyyyy CS zz
Controller to TMP91FY28	End of file record	: 00 0000 01 FF ww
TMP91FY28 to controller	SUM (Upper byte) ((n−1)th byte in Table 3.4.4)	SUM (Upper byte)
TMP91FY28 to controller	SUM (Lower byte) (nth byte in Table 3.4.4)	SUM (Lower byte)
Controller to TMP91FY28	Command ((n+1)th byte in Table 3.4.4)	Next command data

g. Notes on passwords

Passwords can be stored in the address range of 12000H to 4DFFFH. Figure 3.4.3 provides a schematic view of the password area.

1. Password count storage address (PNSA)
   The address specified with PNSA contains the number of passwords (N). A password error occurs in the following cases:
   - PNSA < address 12000H
   - Address 4DFFFH < PNSA
   - N < 8

2. Password comparison start address (PCSA)
   The boot program starts comparing passwords from the address specified with PCSA. The password area to be compared is PCSA to PCSA + N. A password error occurs in the following cases:
   - PCSA < address 12000H
   - Address 4DFFFH < PCSA + N – 1
   - The same data is found in three or more consecutive bytes in the password area. If all bytes in the vector block (4FF00H to 4FFFFH) contain FFH, however, the program assumes that the device is a blank device and does not check the passwords.

3. Password sequence
   The received sequence of passwords is compared with the data stored in the flash memory. A password error occurs in the following case:
   - Received password data does not match the data stored in the corresponding byte in the flash memory.

4. Handling a password error
   If a password error occurs, the boot program stops operation.

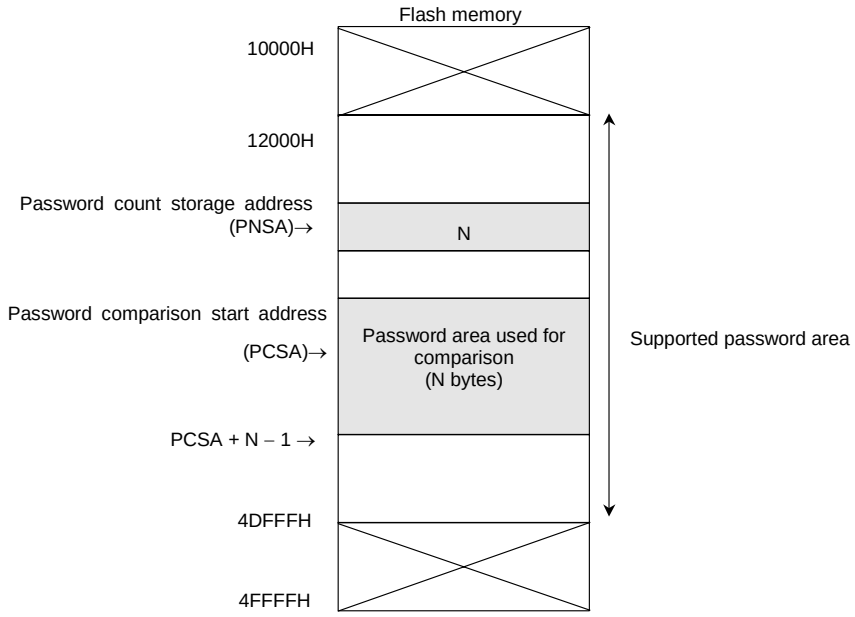


Figure 3.4.3 Schematic View of the Password Area

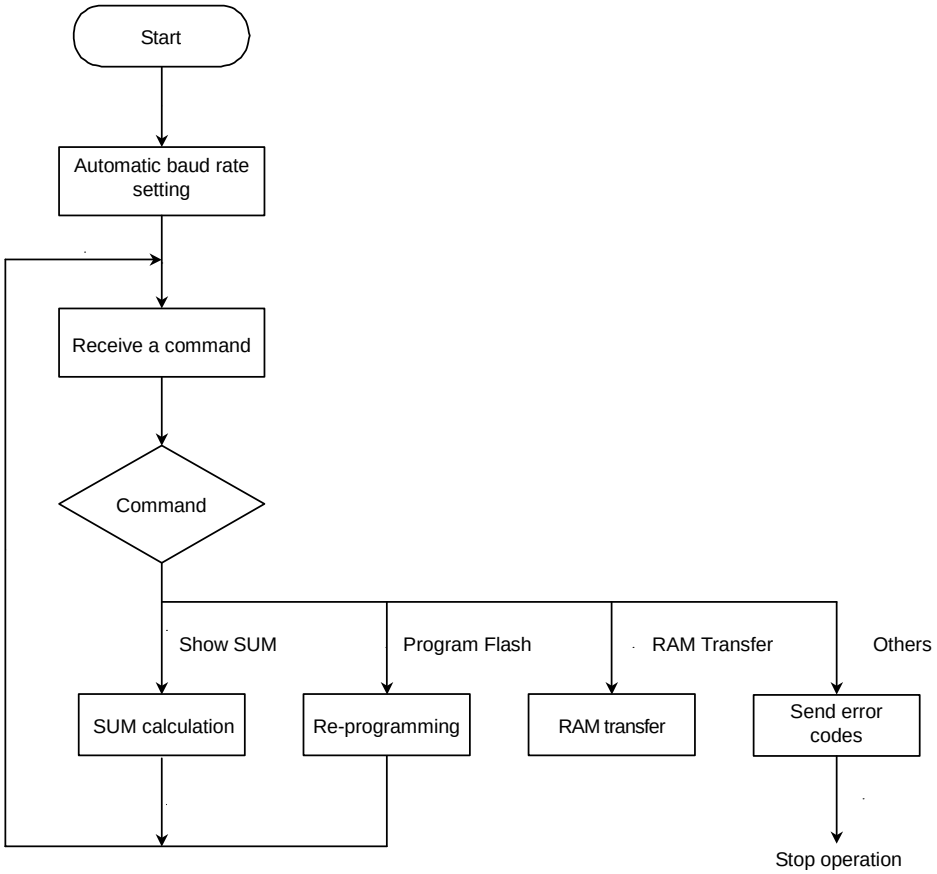


Figure 3.4.4 General Flowchart for Single Boot Mode

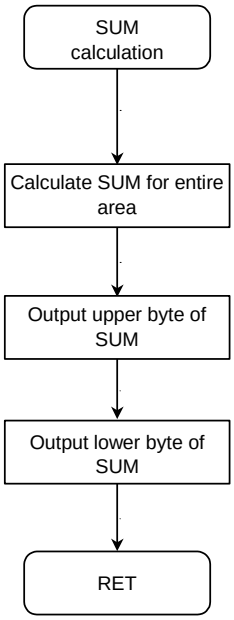


Figure 3.4.5 Show Flash Memory SUM Command

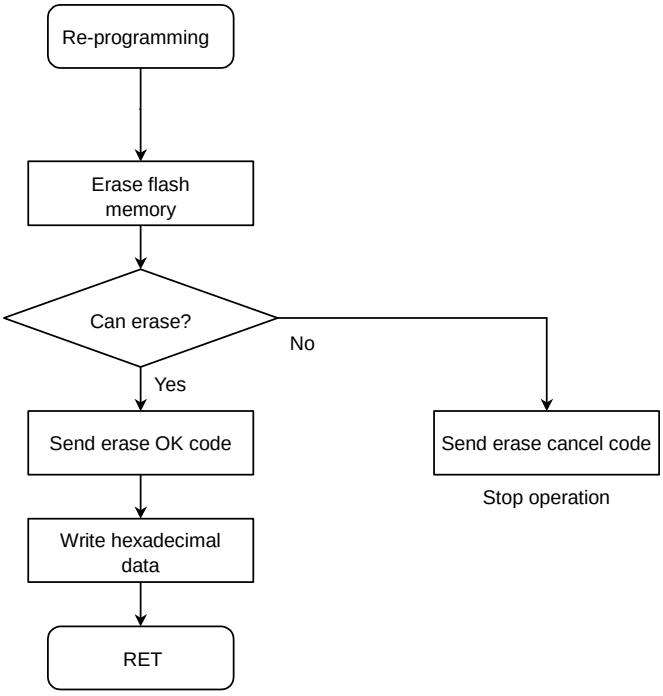


Figure 3.4.6 Program Flash Command

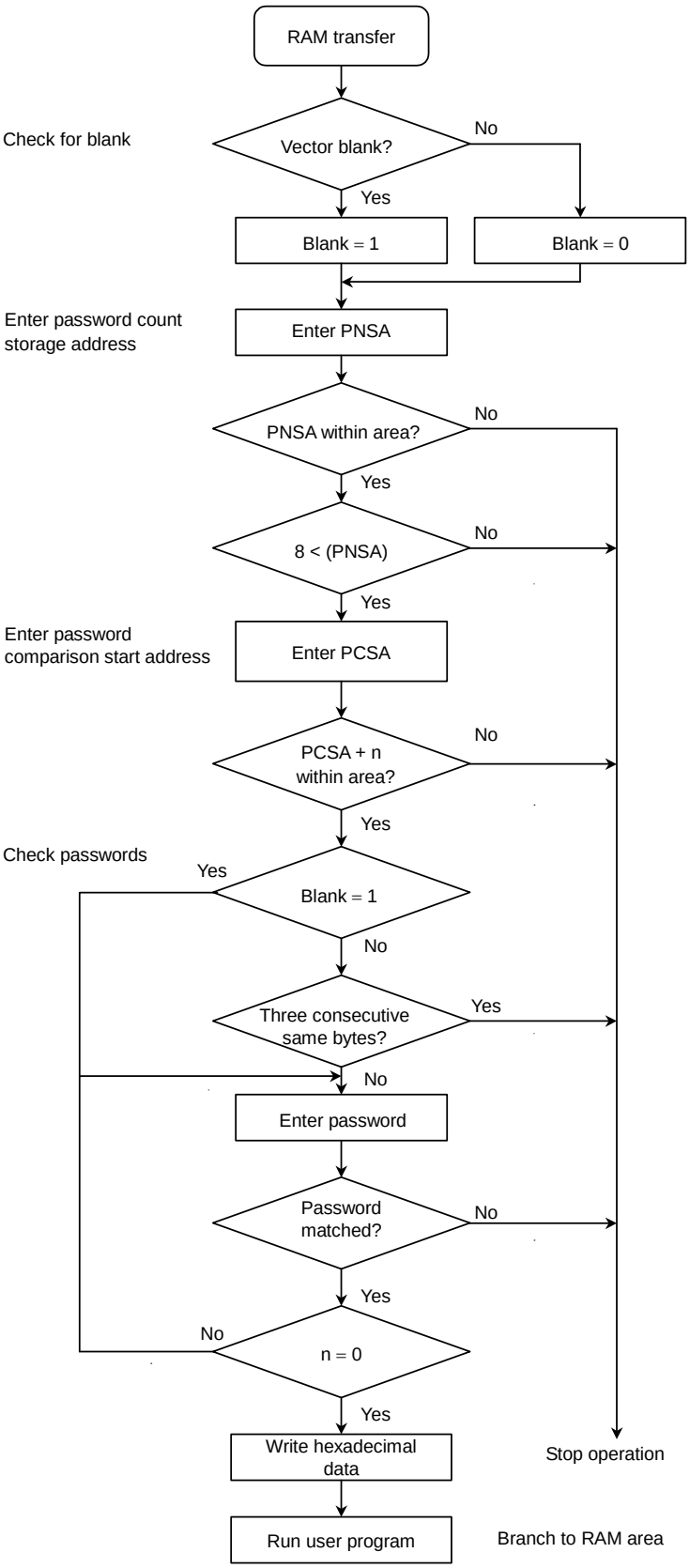


Figure 3.4.7 RAM Transfer Command

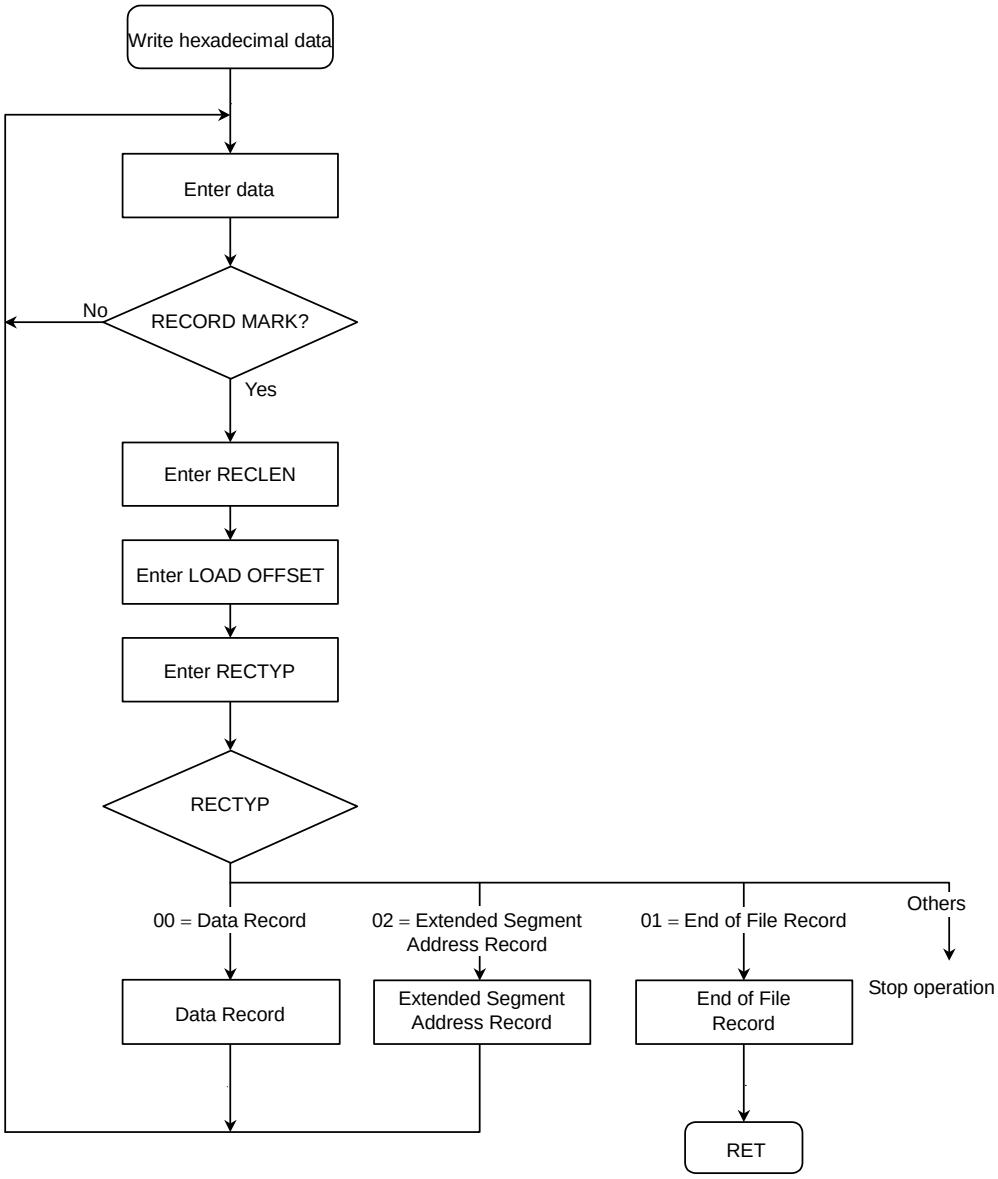


Figure 3.4.8 Writing Hexadecimal Data

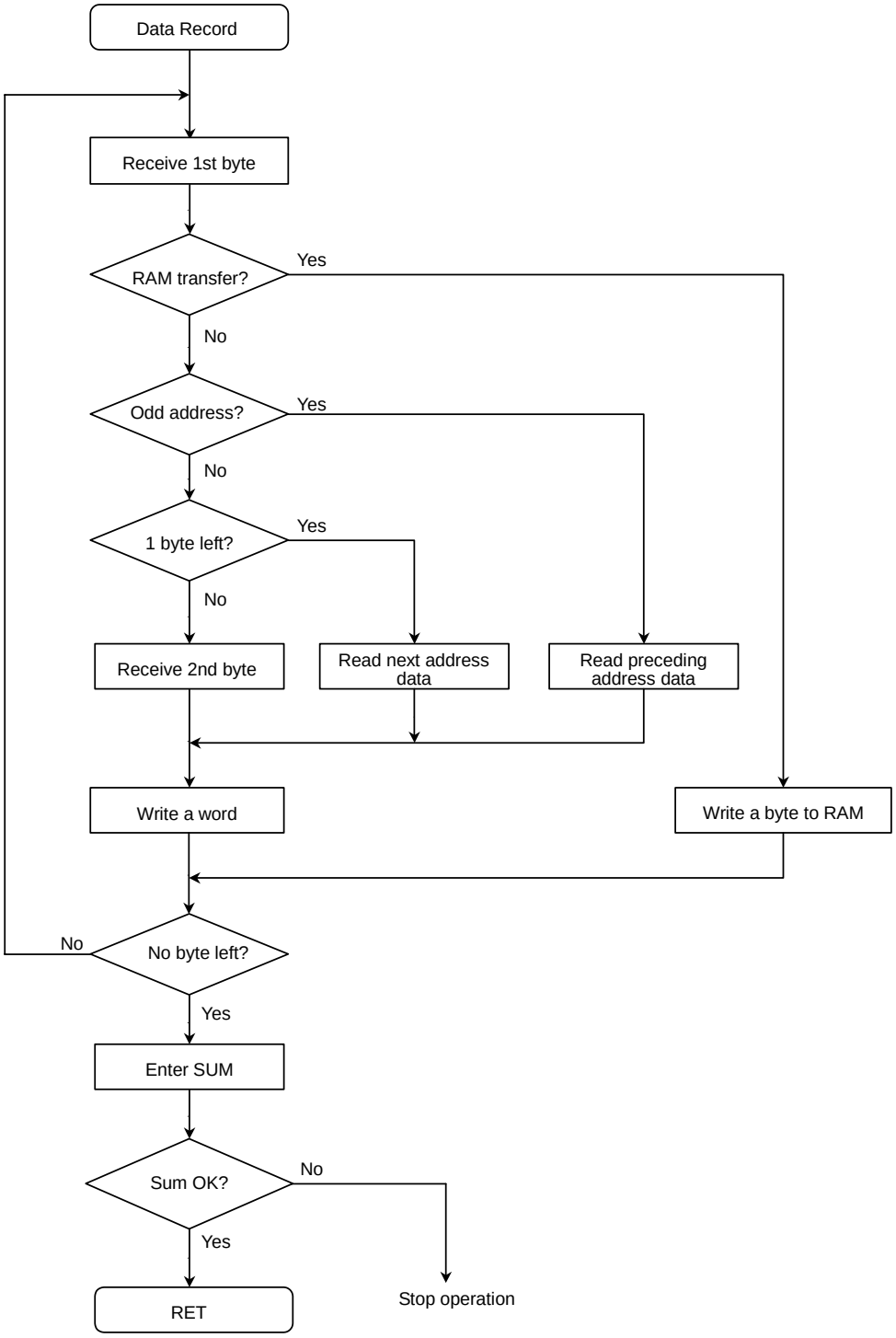


Figure 3.4.9 Data Record

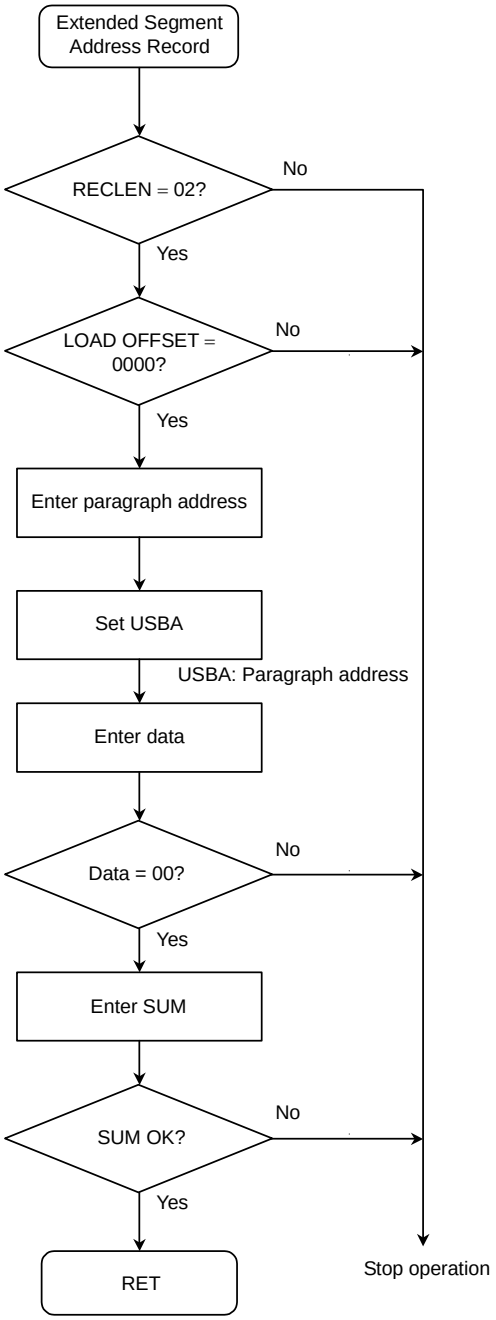


Figure 3.4.10 Extended Segment Address Record

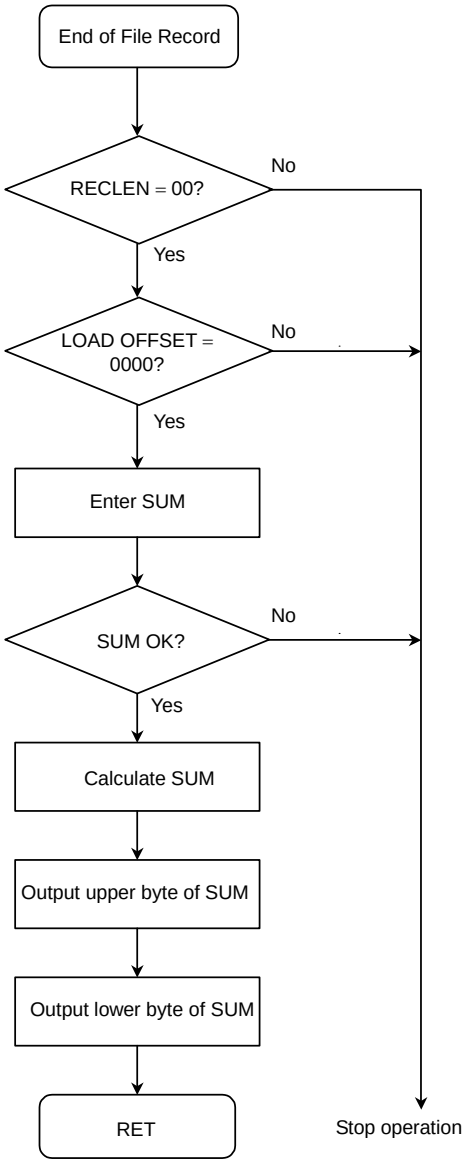


Figure 3.4.11 End of File Record

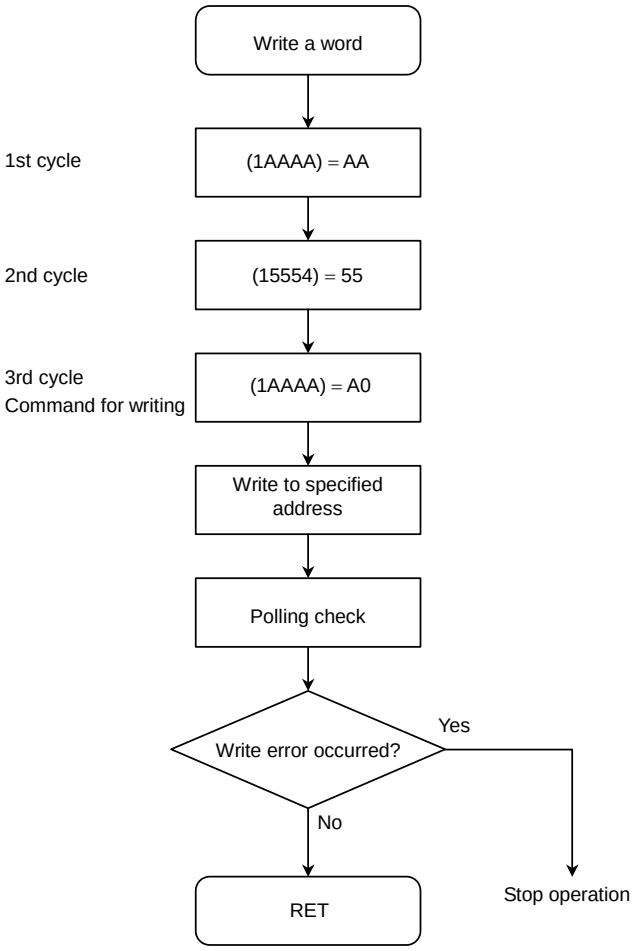


Figure 3.4.12 Writing a Word

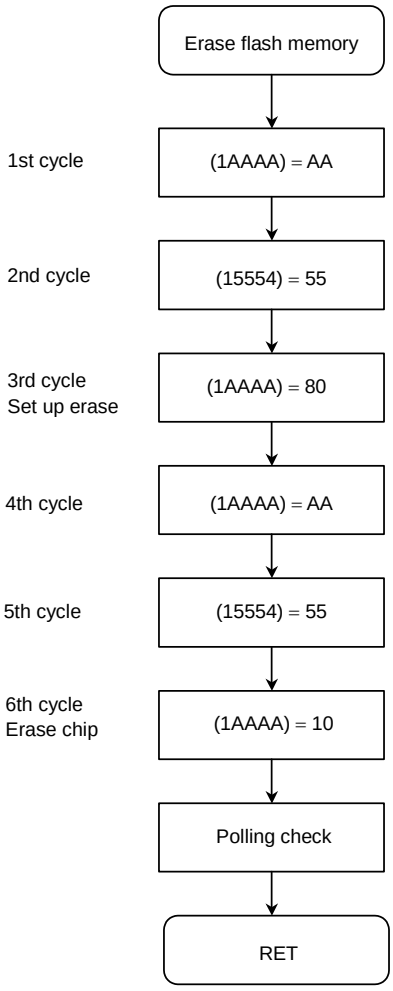


Figure 3.4.13 Erasing the Flash Memory

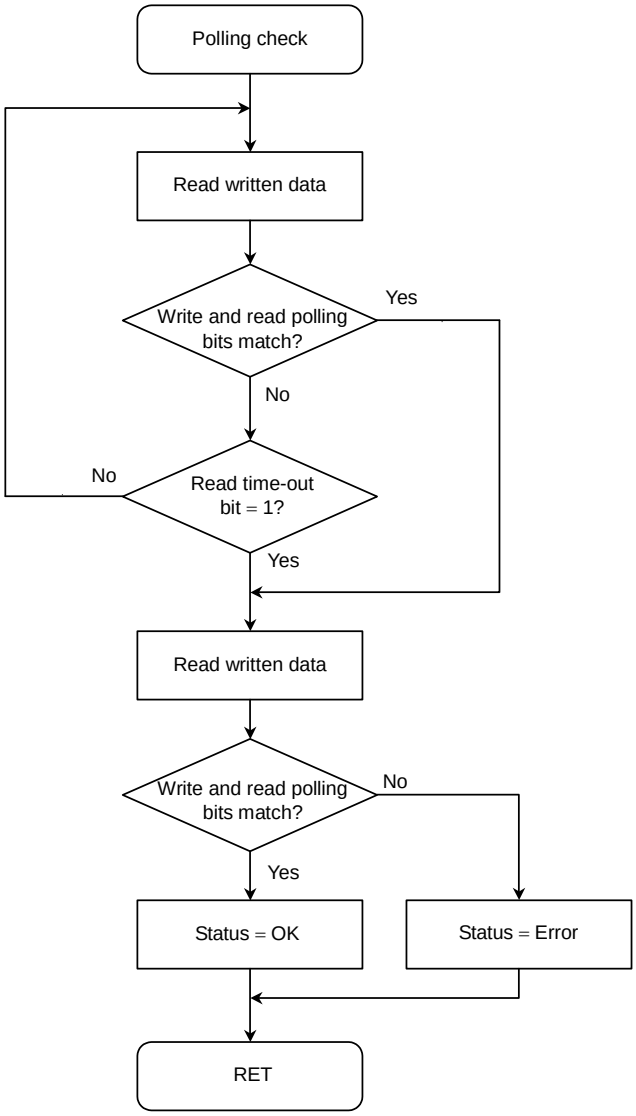


Figure 3.4.14 Data polling

4. Electrical Characteristics (Preliminary)

4.1 Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply voltage	Vcc	−0.5 to 3.0	V
Input voltage	VIN	−0.5 to Vcc + 0.5	
Output current (Per pin)	IOL	2	mA
Output current (Per pin)	IOH	−2	
Output current (Total)	ΣIOL	80	
Output current (Total)	ΣIOH	−80	
Power dissipation (Ta = 85°C)	PD	600	mW
Soldering temperature (10 s)	TSOLDER	260	°C
Storage temperature	TSTG	−55 to 125	
Operating temperature	TOPR	−20 to 70	
Write/erase cycles	N_{EW}	10000	Cycle

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

Point of note about solderability of lead free products (attach “G” to package name)

Test parameter	Test Condition	Note
Solderability	(1) Use of Sn-63Pb solder bath Solder bath temperature = 230°C, Dipping time = 5 [s] The number of times = One, Use of R-type flux	Pass: Solderability rate until forming ≥ 95%
	(2) Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 [s] The number of times = One, Use of R-type flux (use of lead free)	

4.2 DC Electrical Characteristics (1/2)

Parameter		Symbol	Conditions		Min	Typ. (Note)	Max	Unit
Supply Voltage (AVcc = DVcc, AVss = DVss = 0 V)		VCC	fc = 4 to 10 MHz		1.8		2.6	V
Low-level input voltage	P00 to P17 (AD0 to AD15)	VIL	V_{CC} = 1.8 to 2.6 V		−0.3		0.2 Vcc	V
	P20 to P37	VIL1	V_{CC} = 1.8 to 2.6 V				0.2 Vcc	
	$\overline{RESET}$, $\overline{NMI}$, P40 to PA7	VIL2	V_{CC} = 1.8 to 2.6 V				0.15 Vcc	
	AM0 to AM1	VIL3	V_{CC} = 1.8 to 2.6 V				0.3	
	X1	VIL4	V_{CC} = 1.8 to 2.6 V				0.1 Vcc	
High-level input voltage	P00 to P17 (AD0 to AD15)	VIH	V_{CC} = 1.8 to 2.6 V		0.7 Vcc		Vcc + 0.3	V
	P20 to P37	VIH1	V_{CC} = 1.8 to 2.6 V		0.8 Vcc			
	$\overline{RESET}$, $\overline{NMI}$, P40 to PA7	VIH2	V_{CC} = 1.8 to 2.6 V		0.85 Vcc			
	AM0 to AM1	VIH3	V_{CC} = 1.8 to 2.6 V		Vcc − 0.3			
	X1	VIH4	V_{CC} = 1.8 to 2.6 V		0.9 Vcc			
Low-level output voltage		VOL	IOL = 0.4 mA	V_{CC} = 1.8 to 2.6 V			0.15 Vcc	V
High-level output voltage		VOH	IOH = −200 μA	V_{CC} = 1.8 to 2.6 V	0.8 Vcc			

Note: Vcc = 2.0 V, Ta = 25°C, unless otherwise noted.

4.2 DC Electrical Characteristics (2/2)

Parameter	Symbol	Conditions	Min	Typ. (Note 1)	Max	Unit
Input leakage current	ILI	$0.0 \leq V_{IN} \leq Vcc$		0.02	±5	μA
Output leakage current	ILO	$0.2 \leq V_{IN} \leq Vcc - 0.2$		0.05	±10	
Power-down voltage (while RAM is being backed up in STOP mode)	VSTOP	V IL2 = 0.2 Vcc, V IH2 = 0.8 Vcc	1.8		2.6	V
$\overline{RESET}$ pull-up resistor	RRST	V_{CC} = 1.8 to 2.2 V	200		1000	kΩ
		V_{CC} = 2.2 to 2.6 V	100		600	
Pin capacitance	CIO	fc = 1 MHz			10	pF
Schmitt width $\overline{RESET}$, $\overline{NMI}$, P40 to P43, KWI0 to KWI7, P60 to PA7	VTH	V_{CC} = 1.8 to 2.6 V	0.3	0.8		V
Programmable pull-up resistor	RKH	V_{CC} = 1.8 to 2.2 V	200		1000	kΩ
		V_{CC} = 2.2 to 2.6 V	100		600	
NORMAL (Note 2)	Icc	V_{CC} = 1.8 to 2.6 V fc = 10 MHz (Typ.value Vcc = 2.0 V)		10.0	35.0	mA
IDLE2				0.8	1.8	
IDLE1				0.4	1.0	
STOP		V_{CC} = 1.8 to 2.6 V		5	15	μA

Note 1: Vcc = 2.0 V, Ta = 25°C, unless otherwise noted.

Note 2: Test conditions for NORMAL Icc: All blocks operating, output pins open, and input pin levels fixed.

4.3 AC Electrical Characteristics

(1) VCC = 1.8 to 2.6 V

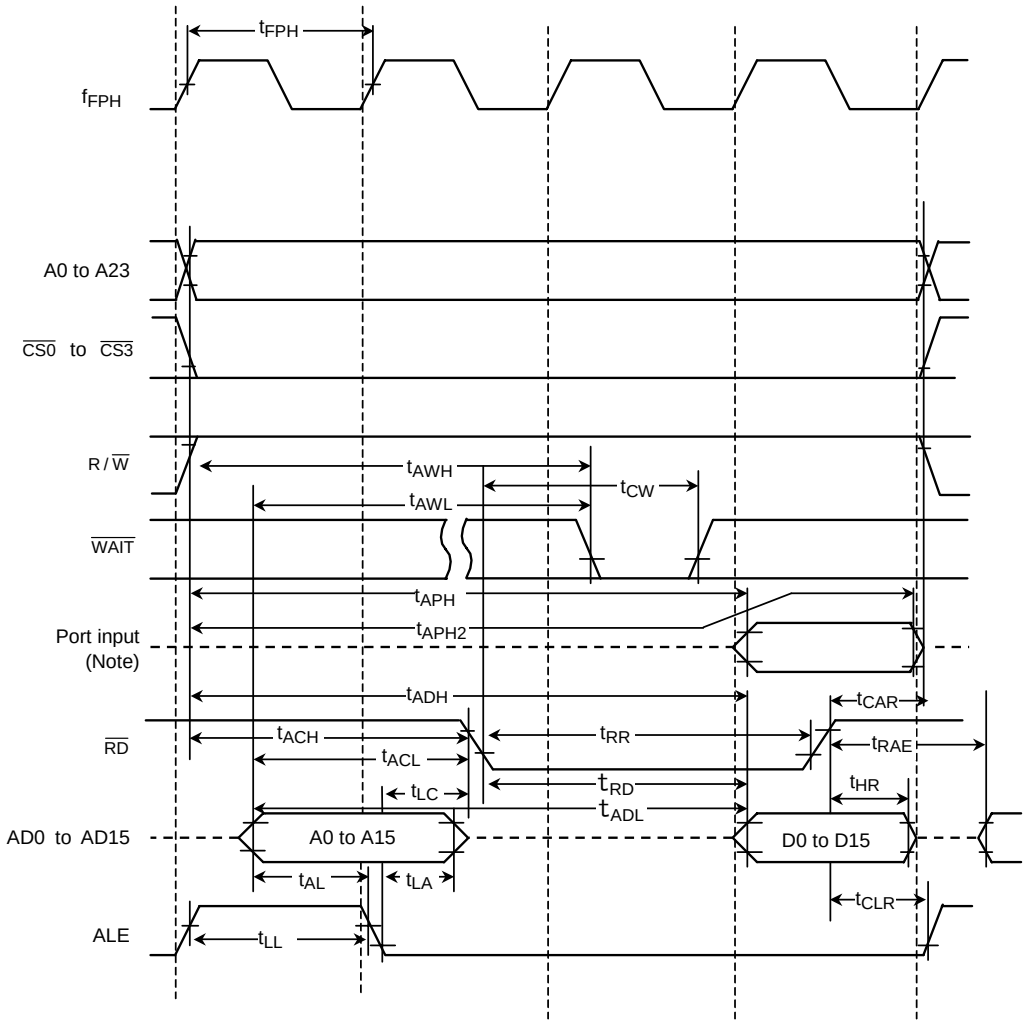
No.	Parameter	Symbol	Equation		f_{FPH} = 10 MHz		Unit
			Min	Max	Min	Max	
1	f_{FPH} cycle period (x)	t_{FPH}	100	250	100		ns
2	A0 to A15 valid to ALE low	t_{AL}	0.5x – 28		22		ns
3	A0 to A15 hold after ALE low	t_{LA}	0.5x – 35		15		ns
4	ALE pulse width high	t_{LL}	x – 40		60		ns
5	ALE low to $\overline{RD}$ or $\overline{WR}$ asserted	t_{LC}	0.5x – 28		22		ns
6	$\overline{RD}$ negated to ALE high	t_{CLR}	0.5x – 20		30		ns
7	$\overline{WR}$ negated to ALE high	t_{CLW}	x – 20		80		ns
8	A0 to A15 valid to $\overline{RD}$ or $\overline{WR}$ asserted	t_{ACL}	x – 75		25		ns
9	A0 to A23 valid to $\overline{RD}$ or $\overline{WR}$ asserted	t_{ACH}	1.5x – 70		80		ns
10	A0 to A23 hold after $\overline{RD}$ negated	t_{CAR}	0.5x – 30		20		ns
11	A0 to A23 hold after $\overline{WR}$ negated	t_{CAW}	x – 30		70		ns
12	A0 to A15 valid to D0 to D15 data in	t_{ADL}		3.0x – 76		224	ns
13	A0 to A23 valid to D0 to D15 data in	t_{ADH}		3.5x – 82		268	ns
14	$\overline{RD}$ asserted to D0 to D15 data in	t_{RD}		2.0x – 60		140	ns
15	$\overline{RD}$ width low	t_{RR}	2.0x – 30		170		ns
16	D0 to D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
17	$\overline{RD}$ negated to next A0 to A15 output	t_{RAE}	x – 30		70		ns
18	$\overline{WR}$ width low	t_{WW}	1.5x – 30		120		ns
19	D0 to D15 valid to $\overline{WR}$ negated	t_{DW}	1.5x – 70		80		ns
20	D0 to D15 hold after $\overline{WR}$ negated	t_{WD}	x – 50		50		ns
21	A0 to A23 valid to $\overline{WAIT}$ input (1 + N) WAIT mode	t_{AWH}		3.5x – 120		230	ns
22	A0 to A15 valid to $\overline{WAIT}$ input (1 + N) WAIT mode	t_{AWL}		3.0x – 100		200	ns
23	$\overline{WAIT}$ hold after $\overline{RD}$ or $\overline{WR}$ asserted	t_{CW}	2.0x + 0		200		ns
24	A0 to A23 valid to port input	t_{APH}		3.5x – 170		180	ns
25	A0 to A23 valid to port hold	t_{APH2}	3.5x		350		ns
26	A0 to A23 valid to port valid	t_{AP}		3.5x + 170		520	ns

AC Measurement Conditions

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

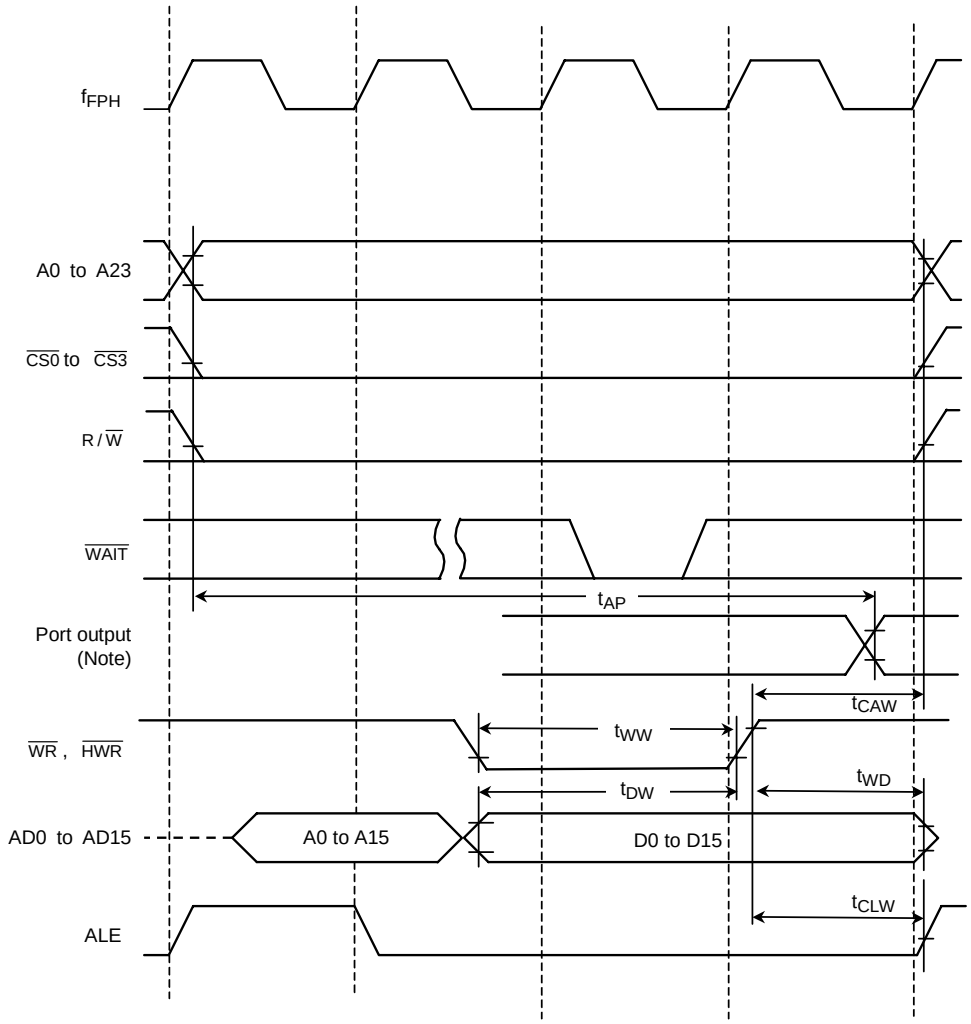
Note: In the table above, the letter x represents the f_{FPH} period, which varies, depending on the programming of the clock gear function. The cycle period of f_{FPH} is half that of the CPU system clock, f_{SYS}.

(2) Read operation timings



Note: Since the CPU accesses the internal area to read data from a port, the control signals of external pins such as $\overline{RD}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

(3) Write operation timings



Note: Since the CPU accesses the internal area to write data to a port, the control signals of external pins such as $\overline{WR}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

4.4 ADC Electrical Characteristics

AVcc = Vcc, AVss = Vss

Parameter		Symbol	Conditions	Min	Typ.	Max	Unit
Analog reference voltage (+)		VREFH	V_{CC} = 1.8 to 2.6 V	Vcc	Vcc	Vcc	V
Analog reference voltage (–)		VREFL	V_{CC} = 1.8 to 2.6 V	Vss	Vss	Vss	
Analog input voltage		VAIN		VREFL		VREFH	
Analog supply current	ADMOD1.VREFON = 1	IREF (VREFL = VSS)	V_{CC} = 1.8 to 2.6 V		0.65	1.0	mA
	ADMOD1.VREFON = 0	IREF (VREFH = VCC)	V_{CC} = 1.8 to 2.6 V		0.02	5.0	μA
Total error (Not including quantization error)		–	V_{CC} = 1.8 to 2.6 V		±1.0	±4.0	LSB

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

Note 2: Minimum operating frequency
Guaranteed when the frequency of the clock selected with the clock gear is 4 MHz or higher with fc used.

Note 3: The supply current flowing through the AV_{CC} pin is included in the digital supply current parameter (I_{CC}).

4.5 SIO Timing (I/O interface mode)

Note: In the tables below, the letter x represents the f_{FPH} period, which varies, depending on the programming of the clock gear function. The cycle period of f_{FPH} is half that of the CPU system clock, f_{SYS}.

(1) SCLK input mode

Parameter	Symbol	Equation		10 MHz (Note)		Unit
		Min	Max	Min	Max	
SCLK period	t_{SCY}	16x		1.6		μs
TXD data to SCLK rise or fall*	t_{OSS}	$t_{SCY}/2 - 4x - 180$ (V_{CC} = 2V ± 10%)		220		ns
TXD data hold after SCLK rise or fall*	t_{OHS}	$t_{SCY}/2 + 2x + 0$		1000		ns
RXD data valid to SCLK rise or fall*	t_{HSR}	3x + 10		310		ns
RXD data valid after SCLK rise or fall*	t_{SRD}		$t_{SCY} - 0$		1600	ns
RXD data hold after SCLK rise or fall*	t_{RDS}	0		0		ns

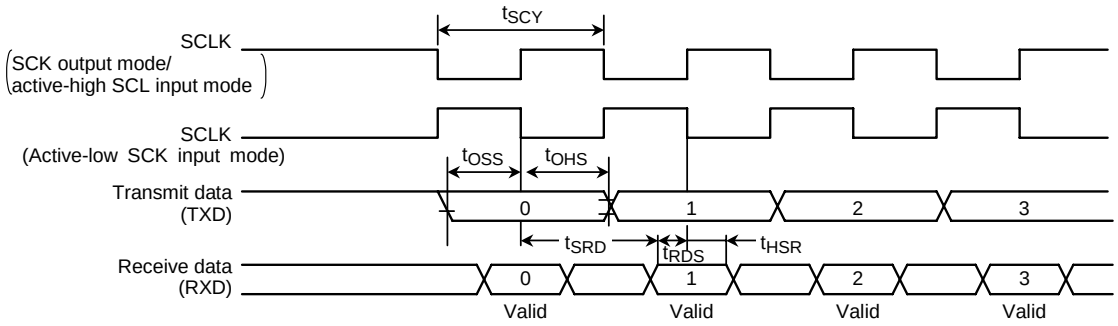
* SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

Note: t_{SCY} = 16x

(2) SCLK output mode

Parameter	Symbol	Equation		10 MHz		Unit
		Min	Max	Min	Max	
SCLK period	t_{SCY}	16x	8192X	1.6	819	μs
TXD data to SCLK rise or fall*	t_{OSS}	$t_{SCY}/2 - 40$		760		ns
TXD data hold after SCLK rise or fall*	t_{OHS}	$t_{SCY}/2 - 40$		760		ns
RXD data valid to SCLK rise or fall*	t_{HSR}	0		0		ns
RXD data valid to SCLK rise or fall*	t_{SRD}		$t_{SCY} - 1X - 180$		1320	ns
RXD data hold after SCLK rise or fall*	t_{RDS}	$1X + 180$		280		ns

* SCLK rise or fall: Measured relative to the programmed active edge of SCLK.



4.6 Event Counters (TA0IN, TB0IN0, TB0IN1, TB1IN0, TB1IN1)

Parameter	Symbol	Equation		10 MHz		Unit
		Min	Max	Min	Max	
Clock cycle period	t_{VCK}	$8x + 100$		900		ns
Clock low pulse width	t_{VCKL}	$4x + 40$		440		ns
Clock high pulse width	t_{VCKH}	$4x + 40$		440		ns

Note: In the table above, the letter x represents the f_{FPH} period, which varies, depending on the programming of the clock gear function. The cycle period of f_{FPH} is half that of the CPU system clock, f_{SYS}.

4.7 Interrupts and Timer Capture

Note: In the tables below, the letter x represents the f_{FPH} period, which varies, depending on the programming of the clock gear function. The cycle period of f_{FPH} is half that of the CPU system clock, f_{SYS}.

(1) $\overline{NMI}$ and INT0 to INT4 interrupts

Parameter	Symbol	Equation		10 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for $\overline{NMI}$ and INT0 to INT4	t_{INTAL}	4X + 40		440		ns
High pulse width for INT0 to INT4	t_{INTAH}	4X + 40		440		ns

(2) INT5 to INT8 interrupts and capture

The input pulse widths for INT5 to INT8 vary with the selected system clock and prescaler clock. The following table shows the pulse widths for different operating clocks:

Selected Prescaler Clock <PRCK1:0>	t_{INTBL} (INT5 to INT8 low pulse width)		t_{INTBL} (INT5 to INT8 high pulse width)		Unit
	Equation	f_{FPH} = 10 MHz	Equation	f_{FPH} = 10 MHz	
	Min	Min	Min	Min	
00 (f_{FPH})	8X + 100	900	8X + 100	900	ns
10 (fc/16)	128Xc + 0.1	12.9	128Xc + 0.1	12.9	μs

Note: Xc indicates the period of the high-speed oscillator clock (fc).

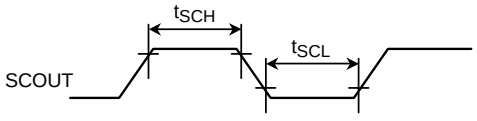
4.8 SCOUT Pin

Parameter	Symbol	Equation		10 MHz		Conditions	Unit
		Min	Max	Min	Max		
Clock high pulse width	t_{SCH}	0.5T − 25		25		V_{CC} = 1.8 to 2.6 V	ns
Clock low pulse width	t_{SCL}	0.5T − 25		25		V_{CC} = 1.8 to 2.6 V	ns

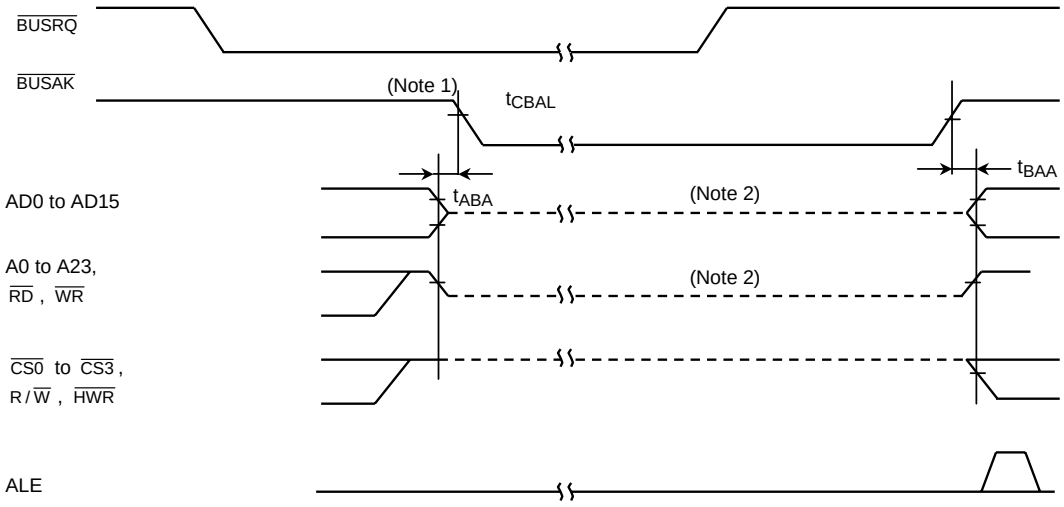
Note: In the table above, the letter T represents the cycle period of the SCOUT output clock.

Measurement condition

- Outpute Levels: High 0.7 Vcc/Low 0.3 Vcc, CL = 10 pF



4.9 Bus Request/Bus Acknowledge



Parameter	Symbol	Equation		f_{FPH} = 10 MHz		Conditions	Unit
		Min	Max	Min	Max		
Bus float to $\overline{BUSAK}$ asserted	t_{ABA}	0	300	0	300	V_{CC} = 1.8 to 2.6 V	ns
Bus float after $\overline{BUSAK}$ negated	t_{BAA}	0	300	0	300	V_{CC} = 1.8 to 2.6 V	ns

Note 1: If the current bus cycle has not terminated due to wait-state insertion, the TMP91FY28 does not respond to $\overline{BUSRQ}$ until the wait state ends.

Note 2: This broken lines indicate that output buffers are disabled, not that the signals are at indeterminate states. The pin holds the last logic value present at that pin before the bus is relinquished. This is dynamically accomplished through external load capacitances. The equipment manufacturer may maintain the bus at a predefined state by means of off-chip resistors, but he or she should design, considering the time (Determined by the CR constant) it takes for a signal to reach a desired state. The on-chip, integrated programmable pull-up/pull-down resistors remain active, depending on internal signal states.

4.10 Recommended Oscillator Circuit

The TMP91FY28 is evaluated by the following resonator manufacturer. The results of evaluation are shown below.

Note: The additional capacitance of the resonator connecting pins are the SUM of load capacitance C1, C2 and the stray capacitance on the target board. Even when recommended constants for C1 and C2 are used, actual load capacitance may vary with the board, possibly resulting in the malfunction of the oscillator. The board should be designed so that the patterns around the oscillator are as short as possible. Toshiba recommends that the resonator be finally evaluated after it is mounted on the target board.

(1) Sample crystal circuit

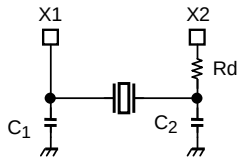


Figure 4.10.1 High-Frequency Oscillator Connection Diagram

(2) Recommended ceramic resonators for the TMP91FY28, manufactured by Murata Manufacturing Co., Ltd.

Ta = –20 to 70°C

Component	Oscillating Frequency [MHz]	Recommended Resonator	Recommended Constants			VCC [V]	Remarks
			C1 [pF]	C2 [pF]	Rd [kΩ]		
High-speed oscillator	4.0	CSTCR4M00G55-R0	(39)	(39)	0	1.8 to 2.6	–
		CSTLS4M00G56-B0	(47)	(47)			
	8.0	CSTCE8M00G52-R0	(10)	(10)			
		CSTLS8M00G53-B0	(15)	(15)			
	10.0	CSTCE10M0G52-R0	(10)	(10)			
		CSTLS10M0G53-B0	(15)	(15)			

- The C1 and C2 constants are enclosed in parentheses for resonator models having built-in capacitors.
- The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL: http://www.murata.co.jp/search/index.html

5. Package Dimensions

P-LQFP100-1414-0.50F

Unit: mm