TOSHIBA

TOSHIBA Original CMOS 32-Bit Microcontroller

TLCS-900/H1 Series

TMP92FD54AIFG

Tentative

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 32-bit Micro-controller

TMP92FD54AIFG

1. Device Outline and Characteristics

The TMP92FD54AI is a high-performance 32-bit microcontroller incorporating a Toshiba-proprietary CPU, the TLCS-900/H1 core. The TMP92FD54AI is developed for various automotive equipments which require high-speed data processing.

Housed in a 100-pin mini-flat package, the TMP92FD54AI is best suited for high-density implementation of user systems.

The characteristics of the TMP92FD54AI are listed below:

(1) Toshiba-proprietary high-speed 32-bit CPU (TLCS-900/H1 CPU)
- Fully-compatible with the instruction codes of the TLCS-900, TLCS-900/L, ELCS-900/L1, TLCS-900/H and TLCS-900/H2
- 16 Mbytes of linear address space
- General-purpose registers and register banks
- Micro DMA: 8 channels (250 ns/4 bytes at fc = 20 MHz)
- Minimum instruction execution time: 50 ns (at fc = 20 MHz)
- Internal data bus: 32-bit wide

(2) Internal memory
- Internal RAM : 32 Kbytes
- Internal ROM : 512-Kbyte flash EEPROM
  3-Kbyte masked ROM (for Single Boot mode)

(3) External memory expansion
- Expandable up to 16-Mbyte (for code and data)
- External data bus: 8-bit wide (The upper address bus is not available when the built-in I/Os are selected.)

(4) Memory controller (MEMC)
- Chip select output: 1 channel

(5) 8-bit timer : 8 channels
- 8-bit interval timer mode (8 channels)
- 16-bit interval timer mode (4 channels)
- 8-bit programmable pulse generation (PPG) output mode (4 channels)
- 8-bit pulse width modulation (PWM) output mode (4 channels)

(6) 16-bit timer : 2 channels
- 16-bit interval timer mode (2 channels)
- 16-bit event counter mode (2 channels)
- 16-bit programmable pulse generation (PPG) output mode (2 channels)
- Frequency measurement mode
- Pulse width measurement mode
- Time difference measurement mode

(7) General-purpose serial interface (SIO) : 2 channels
- I/O interface mode
- Universal asynchronous receiver transmitter (UART) mode

(8) Serial expansion interface (SEI) : 1 channel

(9) Baud rate: 4 M/2 M/500 K bps at fc = 20 MHz

(10) Serial bus interface (SBI): 3 channels
Clock-synchronous 8-bit serial interface mode
I^2C bus mode

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

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

(13) Watch dog timer (WDT)

(14) Timer for real-time clock (RTC)
Can operate with low-frequency oscillator only.

(15) Interrupt controller (INTC) : 60 interrupt sources
9 interrupts from CPU (Software interrupts and undefined instruction interrupt)
42 internal interrupt vectors
9 external interrupt vectors (INT0 to INT7, $\overline{NMI}$)

(16) I/O Port : 68 pins (including multi-function pins)

(17) Standby mode
Four modes: IDLE3, IDLE2, IDLE1 and STOP
STOP mode can be released by 9 external inputs.

(17) Internal voltage detection flag (RAMSTB)

(18) Power supply voltage
V_{CC5} = 4.5 V to 5.25 V
V_{CC3} = 3.3 V (Connect REGOUT (built-in voltage regulator output) to DVCC3.)

(19) Operating temperature: -40 to 85 degree C (0 to 70 degree C when the flash memory is being rewritten.)

(20) Package : LQFP100-P-1414-0.50H

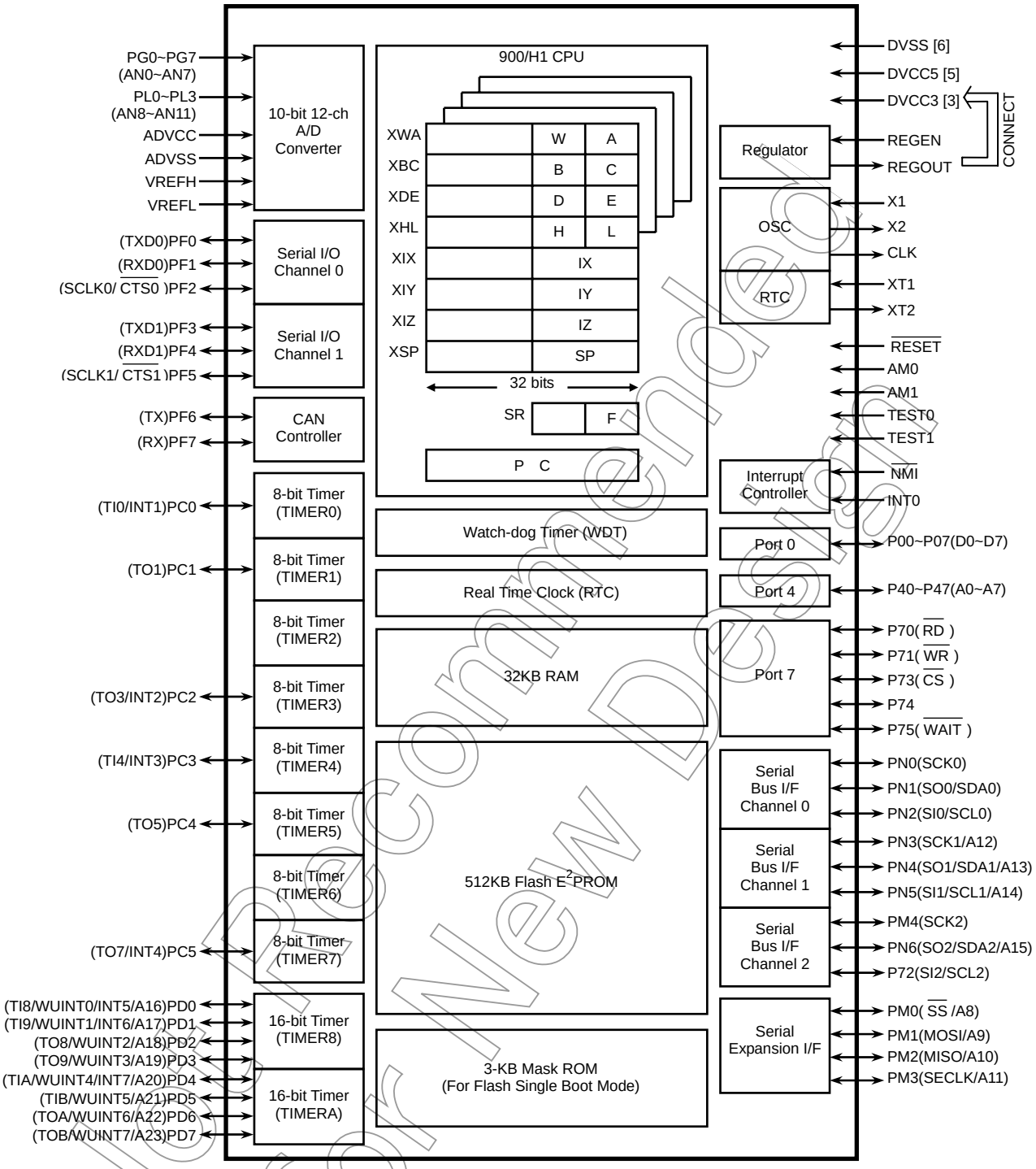


Figure 1.1 TMP92FD54AI Block Diagram

2. Pin Assignment and Functions

2.1 Pin Assignment

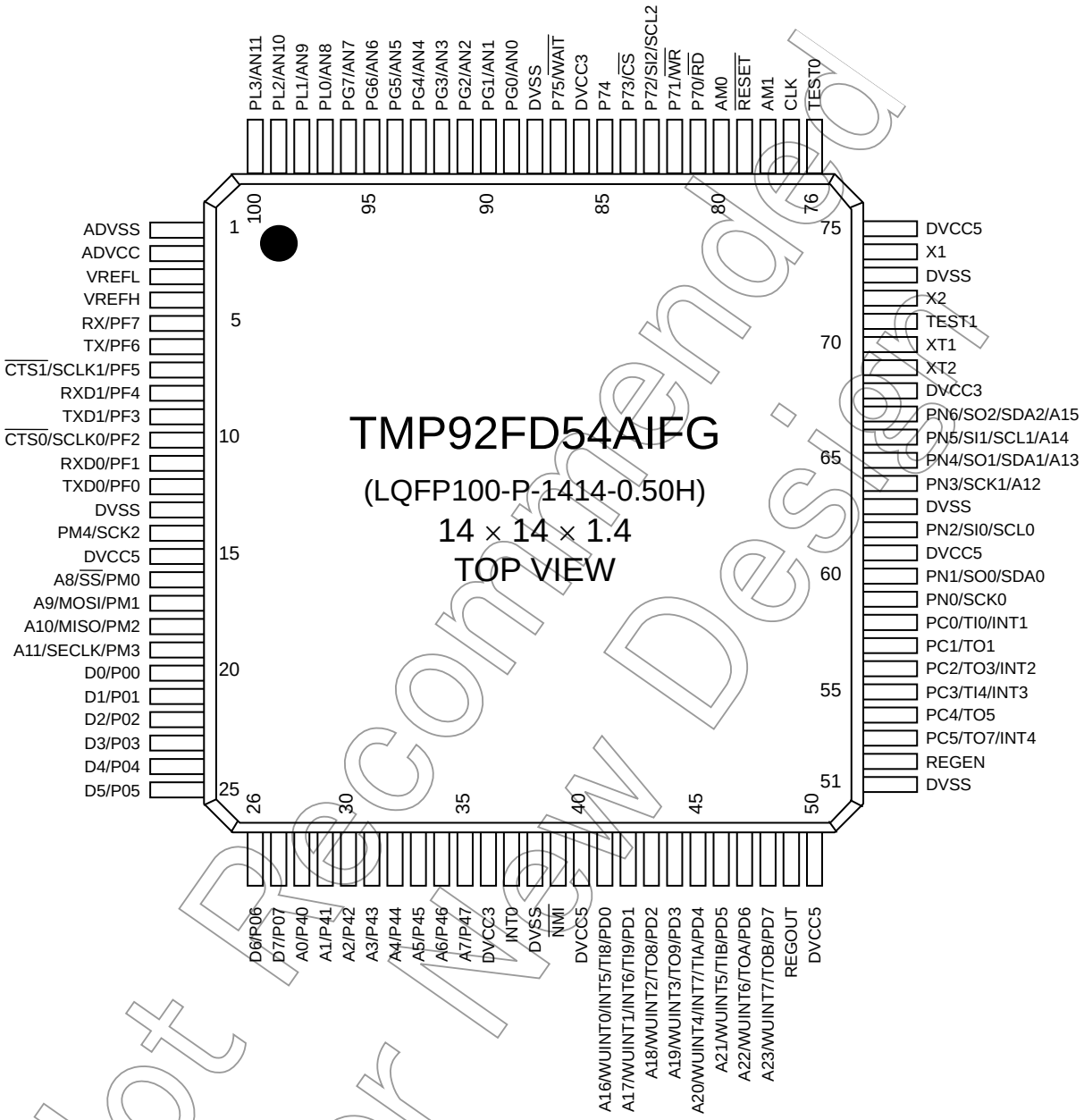


Figure 2.1 Pin Assignment

2.2 Pin Names and Functions

The names and functions of the input/output pins are described in Tables 2.2.1 to 2.2.4.

Table 2.2.1 Input/output pins (1/4)

Pin name	Pin number	Number of pins	In/Out	Function
P00 to P07 D0 to D7	20 to 27	8 (CMOS) (TTL)	in/out in/out	Port 0: I/O port. Input or output specifiable in units of bits. Data: Data bus 0 to 7.
P40 to P47 A0 to A7	28 to 35	8	in/out out	Port4: I/O port. Input or output specifiable in units of bits. Address: Address bus 0 to 7.
P70 $\overline{RD}$	81	1	in/out out	Port70: I/O port. Read: Outputs strobe signal to read external memory.
P71 $\overline{WR}$	82	1	in/out out	Port 71: I/O port. Write: Output strobe signal to write external memory.
P72 SI2 SCL2	83	1	in/out	Port 72: I/O port. SBI channel 2: Input data at SIO mode SBI channel 2: Clock input/output at I²C mode
P73 $\overline{CS}$	84	1	in/out out	Port 73: I/O port. Chip select: Outputs “low” if address is within specified address area.
P74	85	1	in/out	Port 74: I/O port.
P75 $\overline{WAIT}$	87	1	in/out in	Port 75: I/O port. Wait: Signal used to request CPU bus wait.
PC0 TI0 INT1	58	1	in/out in in	Port C0: I/O port. Timer input 0: Input pin for timer 0. Interrupt request pin 1: Rising-edge interrupt request pin.
PC1 TO1	57	1	in/out out	Port C1: I/O port. Timer output 1: Output pin for timer 1.
PC2 TO3 INT2	56	1	in/out out in	Port C2: I/O port. Timer output 3: Output pin for timer 3. Interrupt request pin 2: Rising-edge interrupt request pin.
PC3 TI4 INT3	55	1	in/out in in	Port C3: I/O port. Timer input 4: Input pin for timer 4. Interrupt request pin 3: Rising-edge interrupt request pin.
PC4 TO5	54	1	in/out out	Port C4: I/O port. Timer output 5: Output pin for timer 5.
PC5 TO7 INT4	53	1	in/out out in	Port C5: I/O port. Timer output 7: Output pin for timer 7. Interrupt request pin 4: Rising-edge interrupt request pin.
PD0 TI8 INT5 A16 WUINT0	41	1	in/out in in out in	Port D0: I/O port. Timer input 8: Input pin for timer 8. Interrupt request pin 5: Interrupt request pin with programmable rising/falling edge. Address: Address bus 16. Wake up input 0: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD1 TI9 INT6 A17 WUINT1	42	1	in/out in in out in	Port D1: I/O port. Timer input 9: Input pin for timer 9. Interrupt request pin 6: Rising-edge interrupt request pin. Address: Address bus 17. Wake up input 1: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD2 TO8 A18 WUINT2	43	1	in/out out out in	Port D2: I/O port. Timer output 8: Output pin for timer 8 Address: Address bus 18. Wake up input 2: Wake up request pin with programmable rising, falling or both falling and rising edge.

Table 2.2.2 Input/output pins (2/4)

Pin name	Pin number	Number of pins	In/Out	Function
PD3 TO9 A19 WUINT3	44	1	in/out out out in	Port D3: I/O port. Timer output 9: Output pin for timer 9 Address: Address bus 19. Wake up input 3: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD4 TIA INT7 A20 WUINT4	45	1	in/out in in out in	Port D4: I/O port. Timer input A: Input pin for timer A Interrupt request pin 7: Interrupt request pin with programmable rising/falling edge. Address: Address bus 20. Wake up input 4: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD5 TIB A21 WUINT5	46	1	in/out in out in	Port D5: I/O port. Timer input B: Input pin for timer B. Address: Address bus 21. Wake up input 5: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD6 TOA A22 WUINT6	47	1	in/out out out in	Port D6: I/O port. Timer output A: Output pin for timer A. Address: Address bus 22. Wake up input 6: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD7 TOB A23 WUINT7	48	1	in/out out out in	Port D7: I/O port. Timer output B: Output pin for timer B. Address: Address bus 23. Wake up input 7: Wake up request pin with programmable rising, falling or both falling and rising edge.
PF0 TXD0	12	1	in/out out	Port F0: I/O port. Serial interface channel 0: Transmission data.
PF1 RXD0	11	1	in/out in	Port F1: I/O port. Serial interface channel 0: Receive data.
PF2 SCLK0 $\overline{\text{CTS0}}$	10	1	in/out in/out in	Port F2: I/O port. Serial interface channel 0: Clock input/output. Serial interface channel 0: Data ready to send. (Clear-to-send)
PF3 TXD1	9	1	in/out out	Port F3: I/O port. Serial interface channel 1: Transmission data.
PF4 RXD1	8	1	in/out in	Port F4: I/O port. Serial interface channel 1: Receive data.
PF5 SCLK1 $\overline{\text{CTS1}}$	7	1	in/out in/out in	Port F5: I/O port. Serial interface channel 1: Clock input/output. Serial interface channel 1: Data ready to send. (Clear-to-send)
PF6 TX	6	1	in/out out	Port F6: I/O port. CAN: Transmission data.
PF7 RX	5	1	in/out in	Port F7: I/O port. CAN: Receive data.
PG0 to PG7 AN0 to AN7	89 to 96	8	in in	Port G: Input-only port. Analog input 0 to 7: AD converter input pins.
PL0 to PL3 AN8 to AN11	97 to 100	4	in in	Port L0 to L3: Input-only port. Analog input 8 to 11: AD converter input pins.
PM0 $\overline{\text{SS}}$ A8	16	1	in/out in out	Port M0: I/O port. SEI: Slave select input. Address: Address bus 8.
PM1 MOSI A9	17	1	in/out in/out out	Port M1: I/O port. SEI: Master output, slave input. Address: Address bus 9.

Table 2.2.3 Input/output pins (3/4)

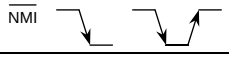
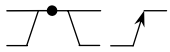
Pin name	Pin number	Number of pins	In/Out	Function
PM2 MISO A10	18	1	in/out in/out out	Port M2: I/O port. SEI: Master input, slave output. Address: Address bus 10.
PM3 SECLK A11	19	1	in/out in/out out	Port M3: I/O port. SEI: Clock input/output. Address: Address bus 11.
PM4 SCK2	14	1	in/out in/out	Port M4: I/O port. SBI channel 2: Clock input/output at SIO mode.
PN0 SCK0	59	1	in/out in/out	Port N0: I/O port. SBI channel 0: Clock input/output at SIO mode.
PN1 SO0 SDA0	60	1	in/out out in/out	Port N1: I/O port. SBI channel 0: Output data input/output at SIO mode SBI channel 0: Data input/output at I²C mode
PN2 SI0 SCL0	62	1	in/out in in/out	Port N2: I/O port. SBI channel 0: Input data at SIO mode SBI channel 0: Clock input/output at I²C mode
PN3 SCK1 A12	64	1	in/out in/out out	Port N3: I/O port. SBI channel 1: Clock input/output at SIO mode Address: Address bus 12.
PN4 SO1 SDA1 A13	65	1	in/out out in/out out	Port N4: I/O port. SBI channel 1: Output data at SIO mode SBI channel 1: Data input/output at I²C mode Address: Address bus 13.
PN5 SI1 SCL1 A14	66	1	in/out in in/out out	Port N5: I/O port. SBI channel 1: Input data at SIO mode SBI channel 1: Clock input/output at I²C mode Address: Address bus 14
PN6 SO2 SDA2 A15	67	1	in/out out	Port N6: I/O port. SBI channel 2: Output data at SIO mode SBI channel 2: data input output at I2C mode Address: Address bus 15.
$\overline{\text{NMI}}$	39	1	in	Non-maskable interrupt: Interrupt request pin with programmable falling or both falling and rising edge. 
INT0	37	1	in	Interrupt request pin 0: Interrupt request pin with programmable level or rising-edge. 
AM0,1	80, 78	2	in	Address Mode selection: Connect AM0 pin to L and AM1 pin to H for Single Chip mode; Connect AM0 and AM1 pins to H for Single Boot mode.
TEST0,1	76, 71	2	in	Test mode pins: Should be tied to GND.
CLK	77	1	out	Programmable clock output (with pull-up resistor)
X1/X2	74, 72	2	in/out	High-frequency oscillator connecting pins: To drive these pins with an external clock, apply clock signals of 3.3 V.
XT1/XT2	70, 69	2	in/out	Low-frequency oscillator connecting pins: To drive these pins with an external clock, apply clock signals of 3.3 V.
$\overline{\text{RESET}}$	79	1	in	Reset: Initializes LSI (with pull-up resistor).
VREFH	4	1	in	AD reference voltage high
VREFL	3	1	in	AD reference voltage low
ADVCC	2	1	-	Power supply pin for AD converter (+5V): Connect the ADVCC pin to 5-V power supply..
ADVSS	1	1	-	GND pin for AD converter: Connect the ADVSS pin to GND (0V).

Table 2.2.4 Input/output pins (4/4)

Pin name	Pin number	Number of pins	In/Out	Function
DVCC5	15, 40, 50, 61, 75	5	-	Power supply pins (+5V): Connect all the DVCC5 pins to 5-V power supply.
DVCC3	36, 68, 86	3	-	Power supply pins (+3.3V): Connect all the DVCC3 pins to REGOUT pin.
DVSS	13, 38, 51, 63, 73, 88	6	-	GND: Connect all DVSS pins to GND (0V).
REGOUT	49	1	out	Regulator output 3.3V: Connect capacitor to stabilize the regulator output.
REGEN	52	1	in	Regulator enable pin: Should be set to H or OPEN (with pull-up resistor).

3. Functional Description of Flash Memory

This chapter describes the structure and operations of the internal flash memory.

The TMP92FD54AI is a microcontroller incorporating a flash memory that is an alternative of the TMP92CD54I's internal ROM. The structural details and functions that are identical with those of the TMP92CD54I are not mentioned in this document. For those details, please refer to the TMP92CD54I datasheet.

3.1 Flash Memory

3.1.1 Features

1) Memory Capacity
The TMP92FD54AI has a 4-Mbit (512-Kbyte) flash memory, which is divided into a total of ten blocks (64 Kbytes × 6 blocks, 56 Kbytes × 2 blocks, 8 Kbytes × 2 blocks) to allow for independent protection from program and erase for each block. When the CPU attempts to access the internal flash memory, it uses the 32-bit data bus.

2) Flash Memory Access
Interleaved access (2-1-1-1 clock access)

3) Program/Erase Time
Chip programming time: 6 seconds (typ.), including program verify operations [30 μs / long word (typ.)]
Chip erase time: 12 seconds (typ.), including program verify operations

Note: These program and erase times are typical values and do not include data transfer overhead. The actual chip program and erase times depend on the programming method used.

4) Programming Procedures
The device has two on-board programming modes: User Boot mode and Single Boot mode. These modes allow a flash memory to be programmed and erased via a serial communication.

- On-Board Programming Modes
  1) User Boot mode (Single-Chip mode)
     Supports use of a user-written programming algorithm.
  2) Single Boot mode
     Downloads new program code using a Toshiba-defined serial interface protocol. User-written programming algorithm is used as the subsequent reprogramming method.

5) Program/Erase Sequence
The flash memory contained in the TMP92FD54AI is compatible with the JEDEC standards, except a few unique functions. Thus, it is easy to change from a discrete flash memory device to the TMP92FD54AI on-chip flash memory. The TMP92FD54AI contains circuitry to perform programming and erase operations automatically. This eliminates the need for the user to code complex program and erase sequences.
The TMP92FD54AI allows the user to protect individual blocks of the flash memory against program or erase through software commands; however, does not supported hardware data protection (12-V VPP).
For more details, please refer to Section 3.5, Flash Memory Program/Erase in On-Board Programming Modes.

Table 3.1.1 Functional Differences from the JEDEC Standards

JEDEC Standards	Changes and Enhancements
• Auto Program • Auto Chip Erase • Auto Block Erase • Data Polling/Toggle Bit	Changed feature: Block protection is available only under software control. Removed feature: Erase Resume/Suspend mode

3.2 Block Diagram

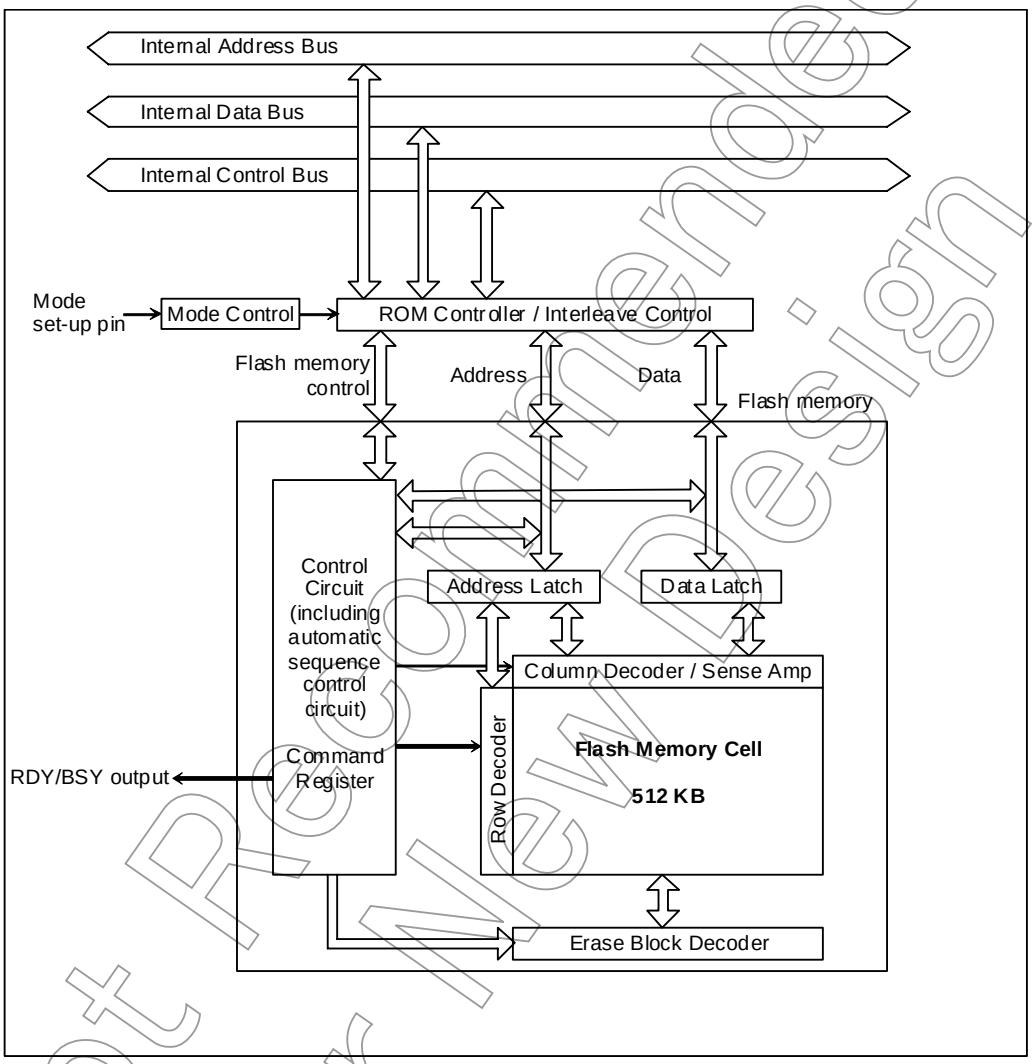


Figure 3.2.1 Flash Memory Block Diagram

3.3 Operating Modes

3.3.1 Overview

The TMP92FD54AI provides two operating modes.

Table 3.3.1 Operating Modes

Operating Mode		Description
Single-Chip Mode		After a reset, the TMP92FD54AI executes out of on-chip flash memory. Single-Chip mode is further divided into Normal mode in which the user application executes and User Boot mode which allows for reprogramming of the flash memory while the TMP92FD54AI is installed on a PC board. The means for selecting between these two modes can be set by the user as desired. For example, the logic state on Port 00 can be used to determine whether to put the flash memory in Normal mode (when Port 00 = 1) or User Boot mode (when Port 00 = 0). The user must include a routine in the application program to handle mode switching.
	Normal Mode	In this mode, the user application program in the flash memory is executed.
	User Boot Mode	This mode is used to rewrite the flash memory while it is installed on a PC board.
Single Boot Mode		After a reset, the TMP92FD54AI executes out of the on-chip boot ROM (which is a mask ROM). The boot ROM contains an algorithm that allows users to transfer the routine for performing on-board programming of the flash memory via a serial port of the TMP92FD54AI to the on-chip RAM. For program/erase operations, the CPU core can issue a command sequence to the flash memory by executing the program/erase routine in the RAM while data from an external host is being received.

The on-chip flash memory can be reprogrammed in one of the following two modes: User Boot mode in Single-Chip mode and Single Boot mode. These two modes that allow the user to program and erase the flash memory in user applications are collectively referred to as on-board programming modes.

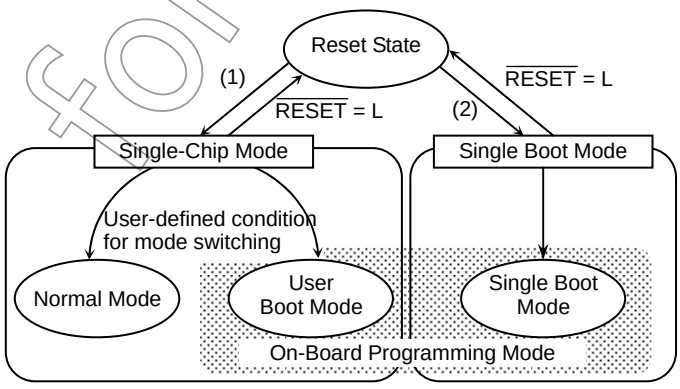
The operating modes for the flash memory, Single-Chip or Single Boot mode, is determined by the logic states on the AM0, AM1, TEST0 and TEST1 pins during a reset sequence.

The CPU starts operations in each operation mode when the Reset state is released after a level input. Do not change a mode state while the CPU is running.

The operating modes and settings, and the mode transitions are shown below.

Table 3.3.2 Operating Modes

	Operating Mode	Input Pins				
		$\overline{\text{RESET}}$	AM1	AM0	TEST1	TEST0
(1)	Single-Chip Mode (Normal and User Boot)	↑	1	0	0	0
(2)	Single Boot Mode		1	1	0	0



Parenthesized numbers, (1) and (2), indicate that the relevant pins are at the logic states shown in Table 3.3.2.

Figure 3.3.1 Mode Transitions

3.3.2 Reset Operation

To reset the TMP92FD54AI, $\overline{\text{RESET}}$ must be asserted for at least 4 μs (at f_{OSC} = 10 MHz) after the internal oscillator and clock multiplier have stabilized while the supply voltage is kept within the normal operating range.

Since the clock multiplier is bypassed during reset, the system clock frequency (fc) is set to 5 MHz (when f_{OSC} = 10 MHz).

For more details, refer to the TMP92CD54I datasheet, Section 3.1.2, Reset Operation.

3.3.3 Memory Maps in Each Operating Mode

The memory map for the TMP92FD54AI varies according to the operation mode. The memory maps, flash memory block architecture and block addresses for each operation mode are shown below.

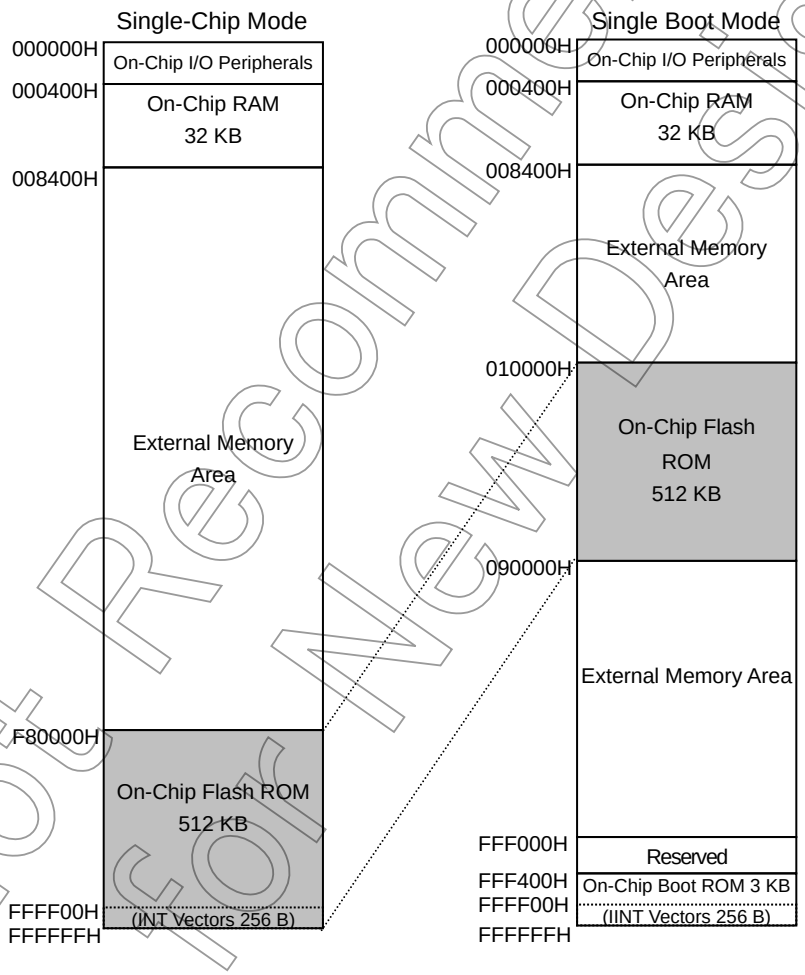


Figure 3.3.2 TMP92FD54AI Memory Maps

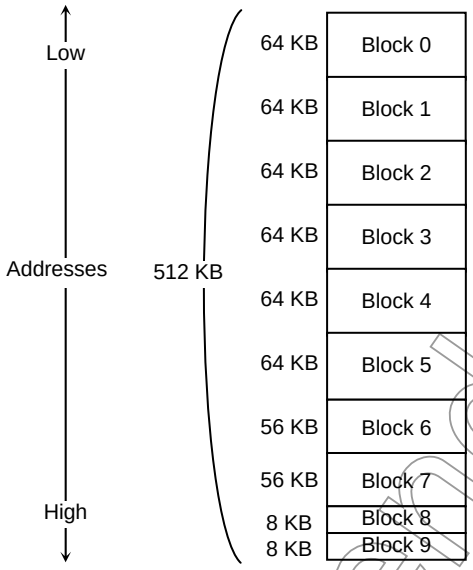


Figure 3.3.3 Flash Memory Block Architecture

Table 3.3.3 Block Addresses

Block	Single-Chip Mode	Single Boot Mode
Block 0 (64 KB)	F80000H - F8FFFFH	010000H - 01FFFFH
Block 1 (64 KB)	F90000H - F9FFFFH	020000H - 02FFFFH
Block 2 (64 KB)	FA0000H - FAFFFFH	030000H - 03FFFFH
Block 3 (64 KB)	FB0000H - FBFFFFH	040000H - 04FFFFH
Block 4 (64 KB)	FC0000H - FCFFFFH	050000H - 05FFFFH
Block 5 (64 KB)	FD0000H - FDFFFFH	060000H - 06FFFFH
Block 6 (56 KB)	FE0000H - FEDFFFH	070000H - 07DFFFH
Block 7 (56 KB)	FEE000H - FFBFFFH	07E000H - 08BFFFH
Block 8 (8 KB)	FFC000H - FFDFFFH	08C000H - 08DFFFH
Block 9 (8 KB)	FFE000H - FFFFFFH	08E000H - 08FFFFH

3.4 On-Board Programming Modes (User Boot and Single Boot Modes)

On-board programming modes allow for reprogramming of the flash memory while the TMP92FD54AI is soldered on a PC board that enables data communications with external devices.

The on-board programming modes include the following:

- User Boot Mode (Supports use of a user-written programming algorithm.)
  This mode allows flash memory reprogramming by executing a user-written algorithm that is preprogrammed in the flash memory for program and erase operations. This reprogramming operation is initiated by user-defined trigger events.
- Single Boot Mode (Supports use of a Toshiba-defined boot program.)
  In this mode, a user-written reprogramming routine is downloaded into the on-chip RAM via a Toshiba-specified communication pin. The boot program offers operation commands, such as RAM Transfer and Erase.

3.4.1 User Boot Mode (in Single-Chip Mode)

User Boot mode allows for flash memory reprogramming by using a user-created reprogramming routine for flash memory. This mode supports situations where the reprogramming routine that is incorporated in the user application code should be used, and where the data transfer should be performed in mode other than UART mode.

This programming algorithm is executed in Single-Chip mode. To reprogram the flash memory, the mode of operation must be switched from Normal mode, in which user application programs usually run, to User Boot mode.

For example, to let the operation mode to be determined on startup, the user application code must include a mode judgment routine as part of the reset procedure so that the operating mode smoothly changes to User Boot mode.

The user must define the conditions for mode switching, based on the logic states on I/O ports of the TMP92FD54AI. Additionally, the user must incorporate a flash memory programming algorithm into the user application code that is to be executed after User Boot mode is entered.

It is impossible to read from the flash memory while it is being erased or programmed; therefore, the programming algorithm must be placed and executed in memory areas that are outside the flash memory, such as on-chip RAM.

Once reprogramming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Normal mode operations.

All interrupts including the nonmaskable interrupt must be globally disabled while the flash memory is being erased or programmed in this mode. When using SIO channels, software can check interrupt request flags in interrupt priority level registers, such as INTES1<ITX1C, IRX1C>, to determine whether reception/transmission has come to an end.

The following sections describe the general procedures for two cases where the programming routine is: 1-A) stored within the TMP92FD54AI flash memory, and 1-B) loaded from an external controller. For a detailed descriptions of the erase and program sequence, refer to Section 3.5, On-Board Programming and Erasure.

3.4.1.1 (1-A) Program/Erase Procedure 1

This procedure is used when the boot program for flash memory programming and erasure is previously included in the on-chip flash memory.

(Step 1) Building a programming environment

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP92FD54AI on a PC board, write the following four program routines into an arbitrary flash block in Single Boot mode.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode
(b) Program/erase routine: Code to download new program code from a host controller and reprogram the flash memory
(c) Copy routine #1: Code to copy routines (a) to (d) to either the on-chip RAM or external memory device.
(d) Copy routine #2: Code to copy routines (a) to (d) contained in either the on-chip RAM or external memory device to the flash memory.

Note: Routine (d) is a code to restore the reprogram routines into the flash memory. It is not necessary if routines (a) to (c) are contained in the flash memory after it is reprogrammed.

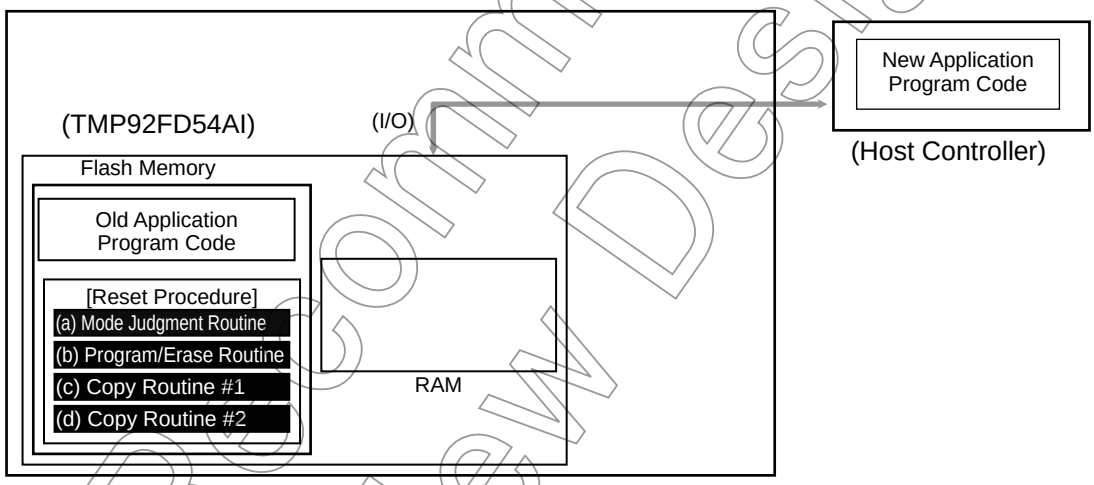


Figure 3.4.1 Building a Programming Environment

(Step 2) Mode Transition to User Boot mode
(when a mode judgment routine is included as part of the reset procedure)

Boot the TMP92FD54AI in Single-Chip mode. After a reset, the reset procedure determines whether to put the TMP92FD54AI flash memory in User Boot mode by calling a mode judgment routine. If mode switching conditions (e.g., pin states) are met, program execution jumps to user boot program for programming and erasure. (All interrupts including non-maskable interrupt must be globally disabled while in User Boot mode.)

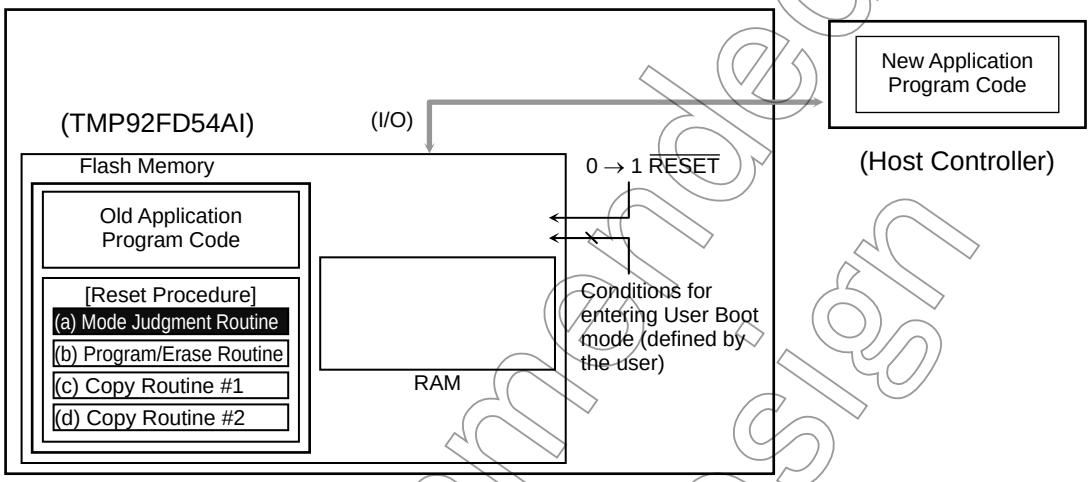


Figure 3.4.2 Mode Transition to User Boot Mode

(Step 3) Copying the user boot routines to the RAM

Execute copy routine #1 to copy the flash programming routines to either the TMP92FD54AI on-chip RAM or an external memory device. (In the following figure, the on-chip RAM is used.)

Note: If routines (a) to (d) are left unerased in the flash memory by using the Auto Block Erase command in Step 4, only the program/erase routine (routine (b)) should be copied into the RAM.

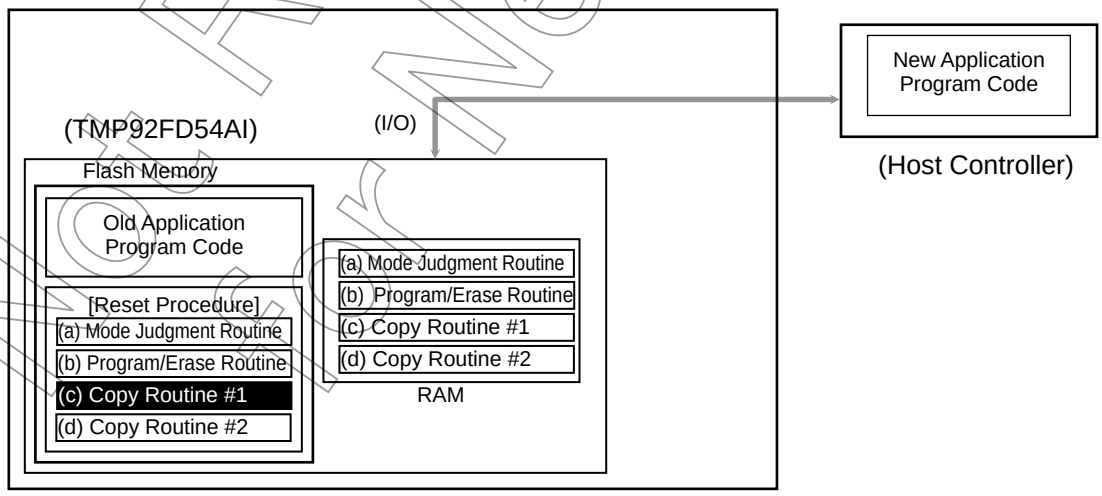


Figure 3.4.3 Copying the User Boot Routines to the RAM

(Step 4) Erasing the flash memory using the program/erase routine

Jump program execution to the flash program/erase routine in the on-chip RAM to erase the flash memory (with the Auto Block Erase or Auto Chip Erase command) containing the old application program code.

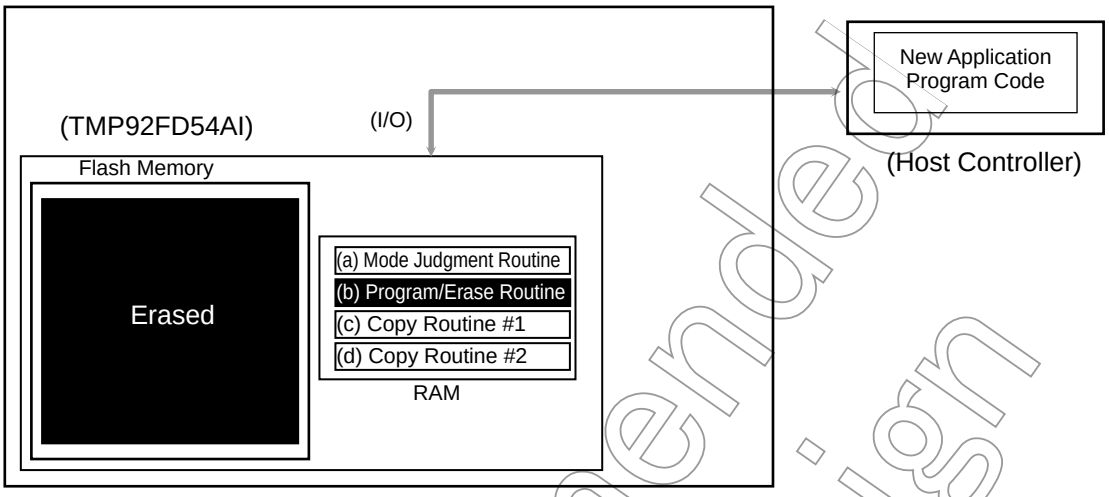


Figure 3.4.4 Erasing the Flash Memory Using the Program/Erase Routine

(Step 5) Restoring the user boot routines to the flash memory

Copy routines (a) to (d) to the flash memory by using copy routine #2 in the on-chip RAM.

Note: Step 5 is not required if routines (a) to (d) are kept in the flash memory by using the Auto Block Erase command.

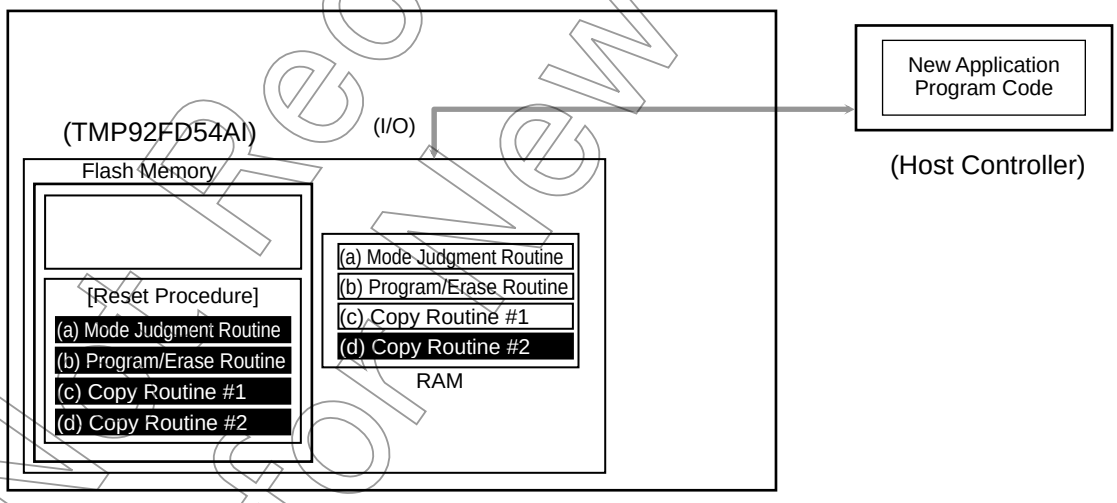


Figure 3.4.5 Restoring the User Boot Routines to the Flash Memory

(Step 6) Writing a new application program code into the flash memory

Execute the flash program/erase routine in the on-chip RAM to download new application program code from the source (host controller) and program it into the erased flash block.

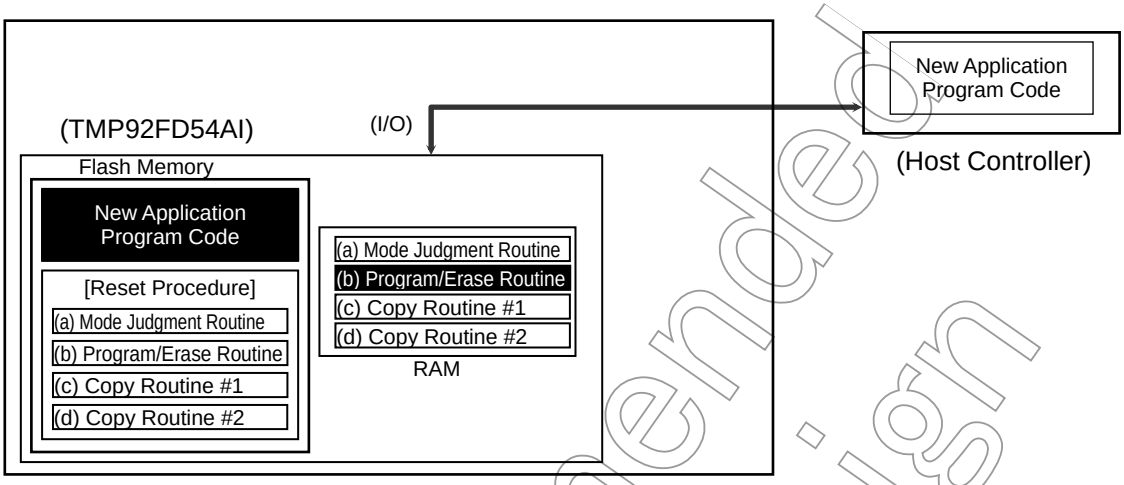


Figure 3.4.6 Writing a New Application Program Code into the Flash Memory

(Step 7) Executing the new application program code

To reset the TMP92FD54AI, drive the $\overline{\text{RESET}}$ pin Low. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.

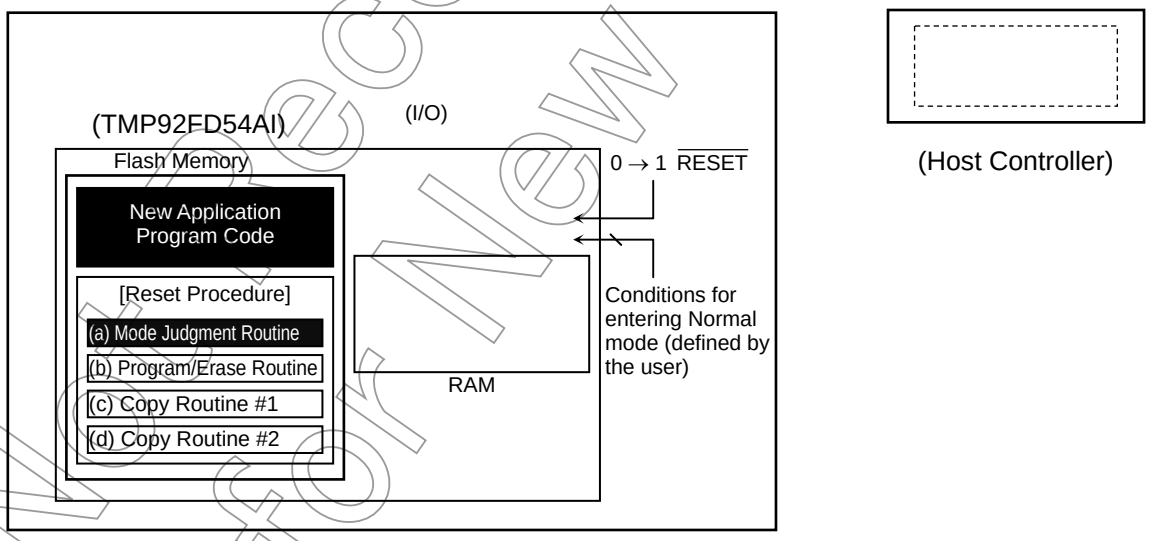


Figure 3.4.7 Executing the New Application Program Code

3.4.1.2 (1-B) Program/Erase Procedure 2

Unlike example (1-A), only the essential part of the boot program codes should be incorporated in the flash memory for this method. Other programs are supplied from a controller as required upon reprogramming the flash memory.

(Step 1) Building a programming environment

Determine the conditions (e.g. pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP92FD54AI on a PC board, write the following two program routines into an arbitrary flash block in Single Boot mode.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode
(b) Transfer routine: Code to download new program/erase code from a host controller

Prepare the following routines on the host controller:

(c) Program/erase routine: Code to reprogram the flash memory
(d) Copy routine #1: Code to copy routines (a) and (b) to the on-chip RAM or external memory device.
(e) Copy routine #2: Code to copy routines (a) and (b) contained in either the on-chip RAM or external memory devices to the flash memory.

Note: Routines (d) and (e) are codes to restore the user boot routines into the flash memory. It is not necessary if routines (a) and (b) are contained in the flash memory after it is reprogrammed.

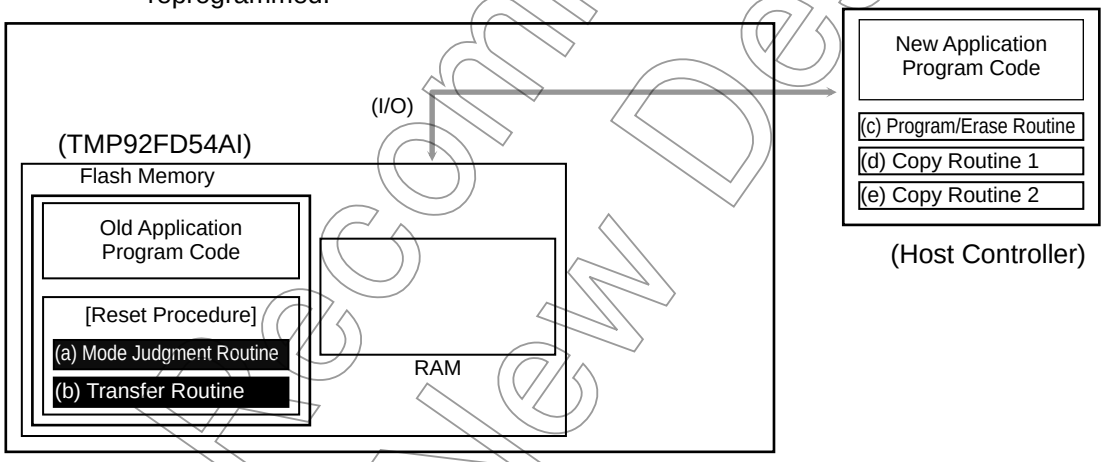


Figure 3.4.8 Building a Programming Environment

(Step 2) Mode Transition to User Boot mode

(when a mode judgment routine is included as part of the reset procedure)

Boot the TMP92FD54AI in Single-Chip mode. After a reset, the reset procedure determines whether to put the TMP92FD54AI flash memory in User Boot mode by calling a mode judgment routine. If mode switching conditions (e.g., pin states) are met, program execution jumps to user boot program for programming and erasure. (All interrupts including non-maskable interrupt must be globally disabled while in User Boot mode.)

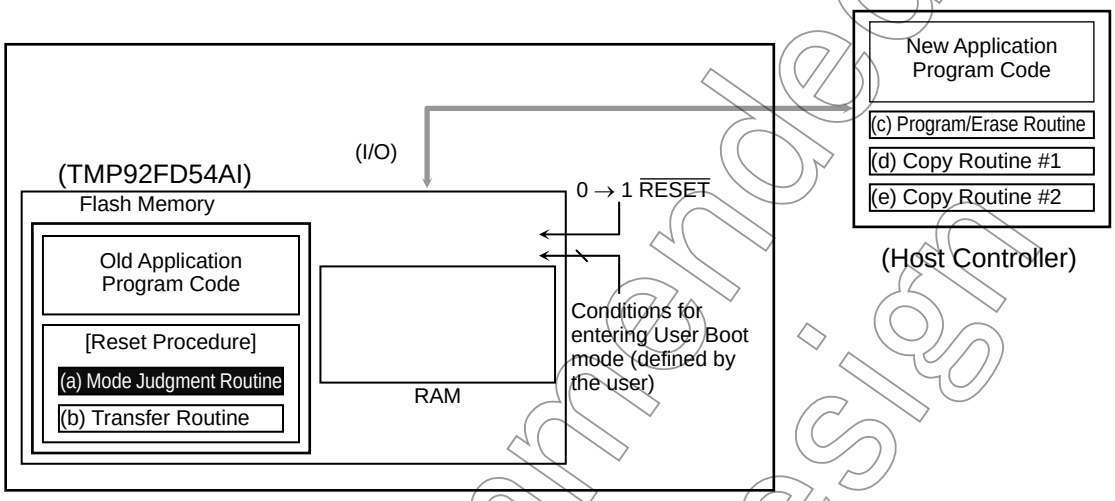


Figure 3.4.9 Mode Transition to User Boot Mode

(Step 3) Copying the user boot routines from the host controller to the RAM

Execute the transfer routine (routine (b)) to download routines (c) to (e) from the source (host controller) to the on-chip RAM or an external memory device. (In the following figure, the on-chip RAM is used.)

Note: If routines (a) and (b) are left unerased in the flash memory by using the Auto Block Erase command in Step 5, only the program/erase routine (routine (c)) should be copied into the on-chip RAM.

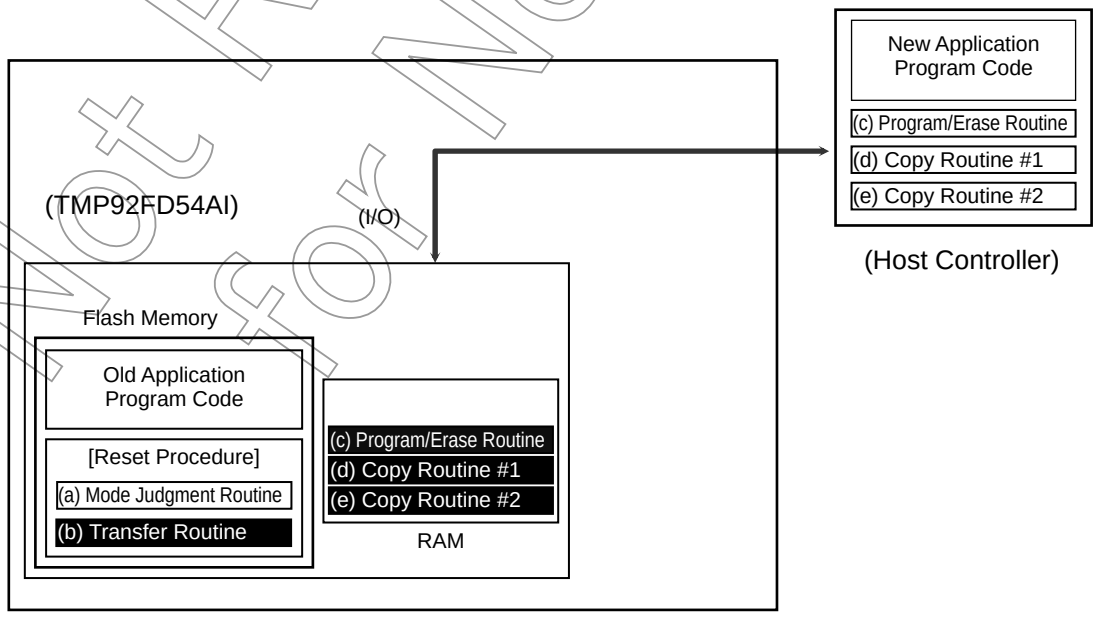


Figure 3.4.10 Copying the User Boot Routines in User Boot Mode (from the Host Controller to the RAM)

(Step 4) Copying the user boot routines from the flash memory to the on-chip RAM

Program execution jumps to the routine in the on-chip RAM. Execute copy routine #1 to copy routines (a) and (b) to the on-chip RAM.

Note: Step 4 is not required if routines (a) and (b) are kept in the flash memory by using the Auto Block Erase command in Step 5.

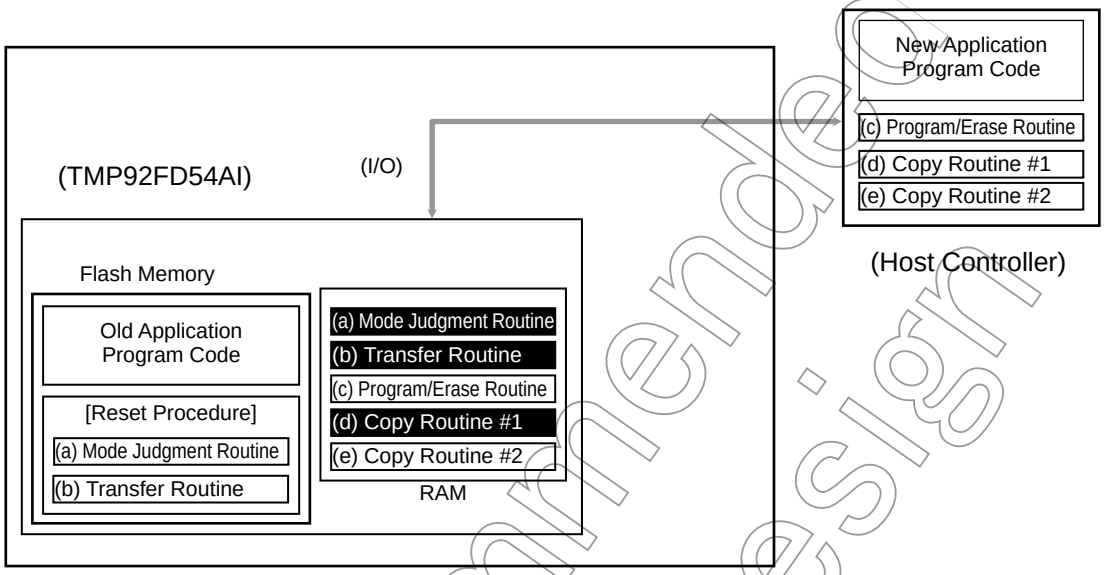


Figure 3.4.11 Copying the User Boot Routines (from the Flash Memory to the On-Chip RAM)

(Step 5) Erasing the flash memory using the program/erase routine

Execute the program/erase routine (routine (c)) to erase a flash block or chip containing the old application program code.

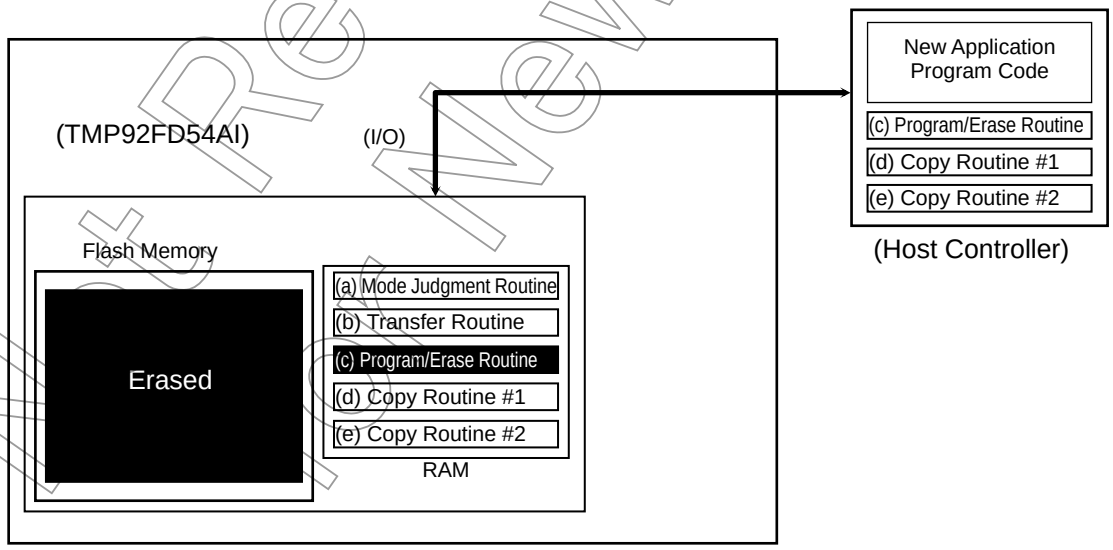


Figure 3.4.12 Erasing the Flash Memory Using the Program/Erase Routine

(Step 6) Restoring the user boot routines to the flash memory

Execute copy routine #2 to copy routines (a) and (b) in the on-chip RAM to the flash memory.

Note: Step 6 is not required if routines (a) and (b) are kept in the flash memory by using the Auto Block Erase command in Step 5.

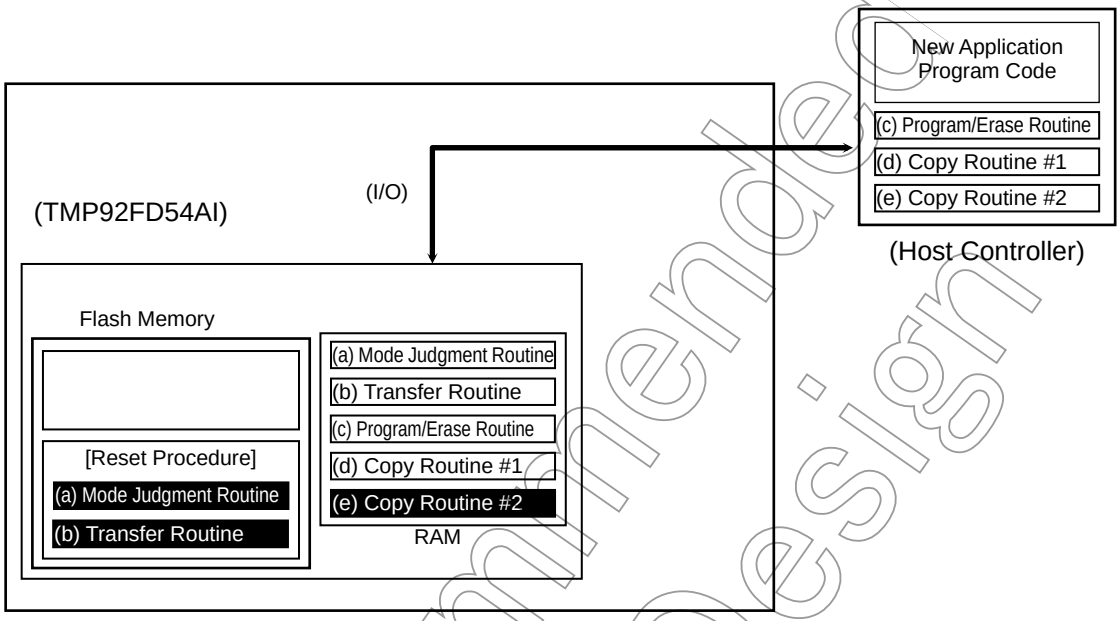


Figure 3.4.13 Restoring the User Boot Routines to the Flash Memory

(Step 7) Writing a new application program code into the flash memory

Execute the flash program/erase routine (c) in the on-chip RAM to download new application program code from the source (host controller) and program it into the erased flash block.

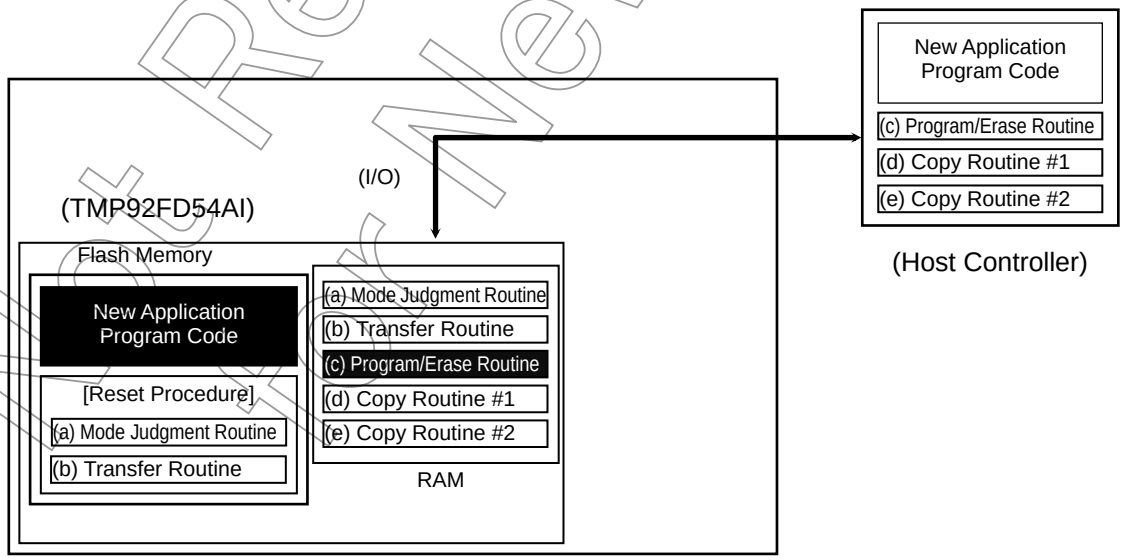


Figure 3.4.14 Writing a New Application Program Code into the Flash Memory

(Step 8) Executing the new application program code

To reset the TMP92FD54AI, drive the $\overline{\text{RESET}}$ pin Low. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.

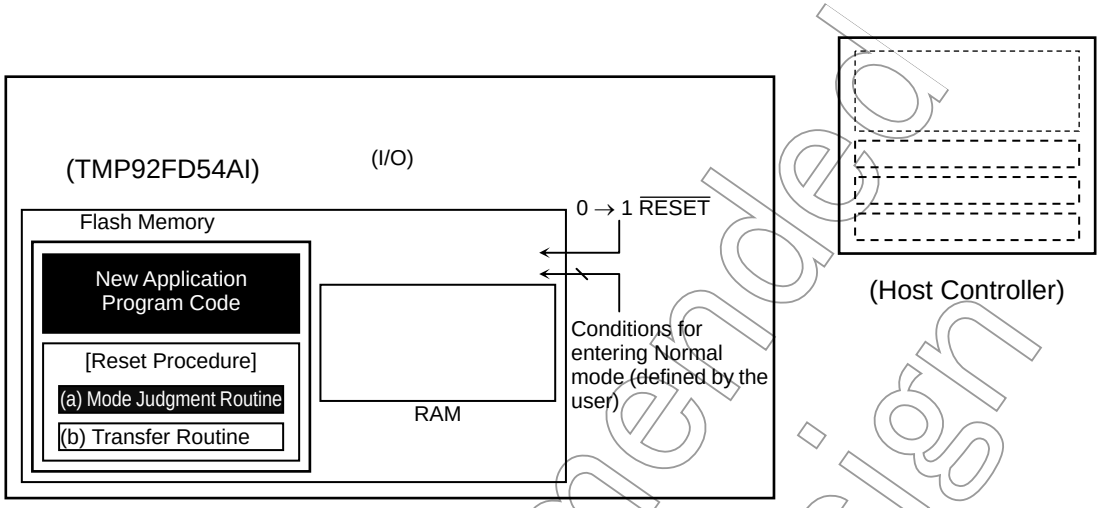


Figure 3.4.15 Executing the New Application Program Code

3.4.2 Single Boot Mode

In Single Boot mode, the flash memory can be programmed and erased by using a program contained in the TMP92FD54AI on-chip boot ROM. This boot ROM is a masked ROM. The flash memory is programmed and erased by executing a program/erase routine (a user-created reprogramming routine) that is transferred to the RAM from an external host controller. In this mode, the on-chip boot ROM is mapped to the address area including the interrupt vector table for executing the on-chip boot ROM code. The flash memory is mapped to an address space different from the boot ROM area (refer to Figure 3.3.2).

The SIO (SIO1) of the TMP92FD54AI is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMP92FD54AI on-chip RAM. Then, the flash memory is reprogrammed by executing the programming routine.

The host sends out both commands and programming data to reprogram the flash memory. Communications between the SIO1 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is checked before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted.

As in the case of User Boot (Single-Chip) mode, all interrupts including nonmaskable interrupt must be globally disabled in Single Boot mode while the flash memory is being erased or programmed. When using SIO channels, software can check interrupt request flags in interrupt priority level registers, such as INTES1<ITX1C, IRX1C>, to determine whether reception/transmission has come to an end.

Once reprogramming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Normal mode operations. For a detailed description of the program/erase sequence, refer to Section 3.5, Flash Memory Program/Erase in On-Board Programming Modes.

Note: Don't change the pin states of TEST0, TEST1, AM0 and AM1 during a reprogramming operation of the flash memory.

3.4.2.1 Using the Program/Erase Routine in the On-Chip Boot ROM

(Step 1) Building a programming environment

Since a program/erase routine and programming data are transferred via the SIO1, the SIO1 must be connected to a host controller. Prepare a program/erase routine (routine (a)) on the host controller.

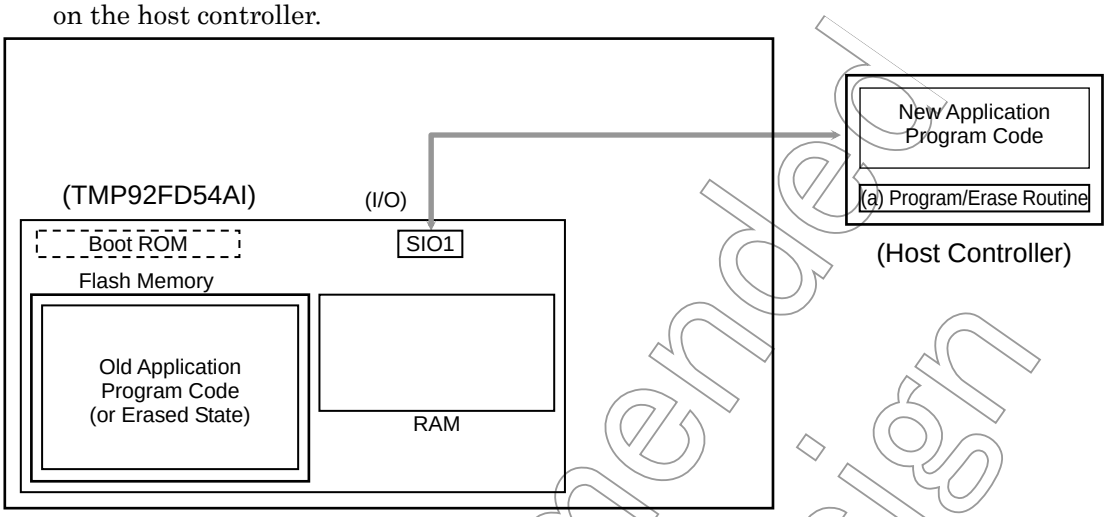


Figure 3.4.16 Building a Programming Environment

(Step 2) Rebooting in Single Boot mode and the password matching
(Rebooting from the on-chip ROM)

Reset the TMP92FD54AI with the mode setting pins held at appropriate logic values so that the CPU reboots from the on-chip boot ROM. To transfer the program/erase routine (routine (a)) from the source (host controller) to the on-chip RAM via the SIO, communications are made following the prescribed protocol shown in Table 3.4.4. The password transferred from the host controller is first compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is a sequence of 12 0xFF bytes.) For more details on password matching, see Section 3.4.2.14, Password.

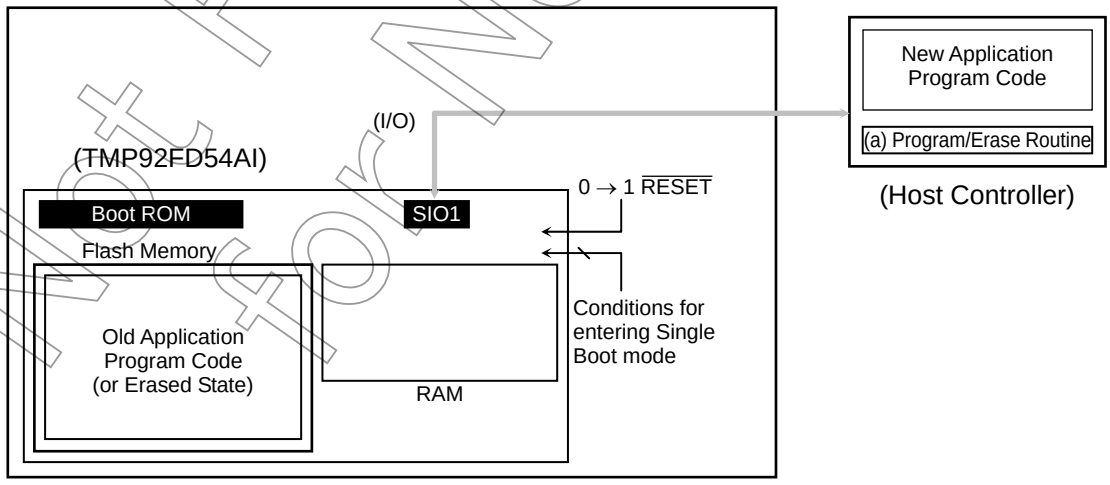


Figure 3.4.17 Rebooting in Single Boot Mode (Rebooting from the on-chip ROM)

(Step 3) Copying the program/erase routine from the host controller to the on-chip RAM

If the password was correct, the boot ROM downloads, via the serial link (SIO1), the program/erase routine (routine (a)) from the host controller into the on-chip RAM. The program/erase routine must be stored in the address range between 000400H and 006BFFH.

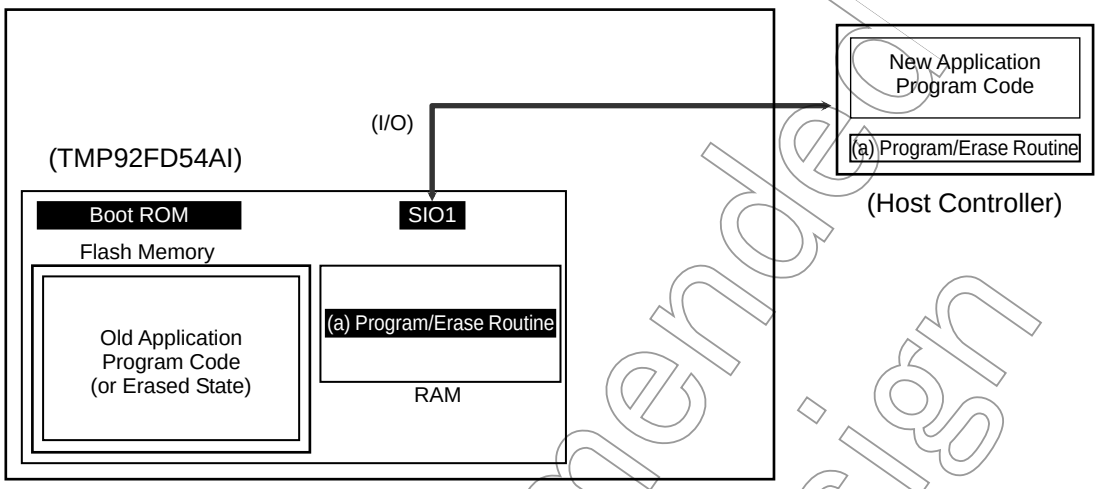


Figure 3.4.18 Copying the Program/Erase Routine from the Host Controller to the On-Chip RAM

(Step 4) Erasing the flash memory by executing the program/erase routine

The CPU jumps to the program/erase routine in the on-chip RAM to erase the flash block containing the old application program code as required. (The Auto Block Erase or Auto Chip Erase command should be used.)

Note: Since the boot program has the Auto Chip Erase & Unprotect commands, the host controller is capable of erasing the flash chip without using the program/erase routine. To perform other operations such as the block erase and block protect operations, corresponding codes should be included in the program/erase routine.

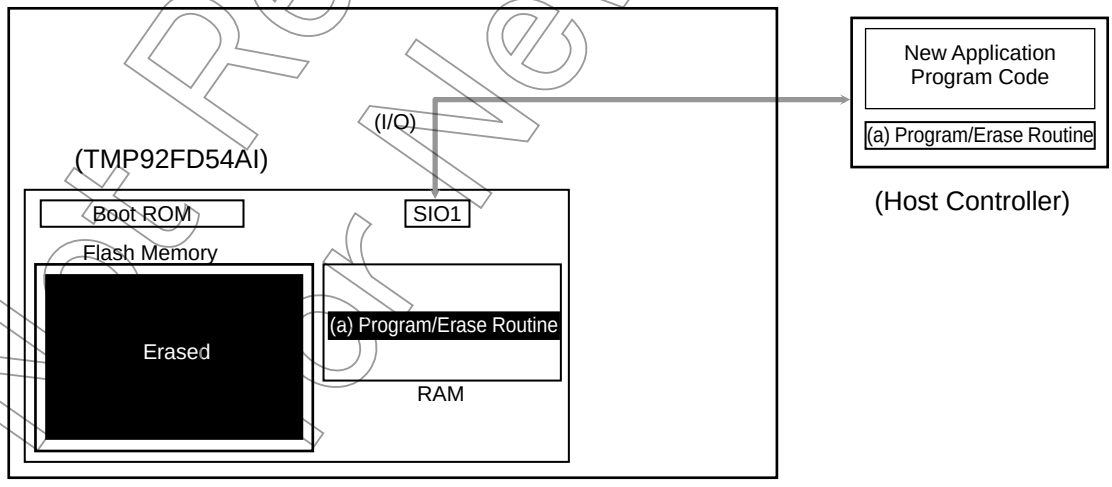


Figure 3.4.19 Erasing the Flash Memory by Executing the Program/Erase Routine

(Step 5) Copying a new application program code

Next, the program/erase routine (routine (a)) in the on-chip RAM downloads a new application program code from the source (host controller) and programs it into the erased flash block. Once programming is complete, protection of that flash block is turned on as required.

In the example below, a programming data transfer is performed using the same SIO communication format as for the program/erase routine. However, once the program/erase routine has been transferred, it is free to change the setting, such as the transfer path and the source of the transfer. Create board hardware and a program/erase routine to suit your particular needs.

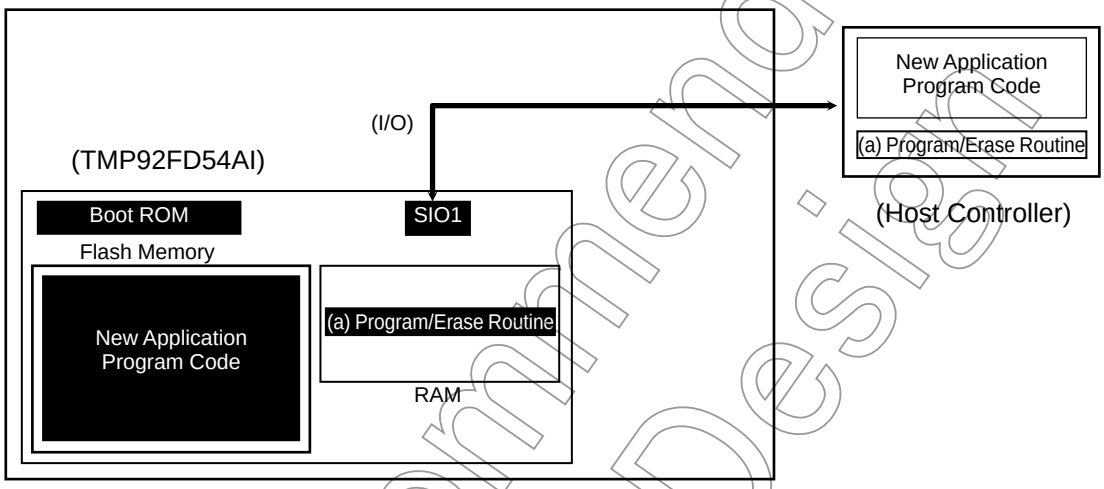


Figure 3.4.20 Copying a New Application Program Code

(Step 6) Executing the new application program

When programming of the flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the TMP92FD54AI reboots in Single-Chip mode to execute the new program.

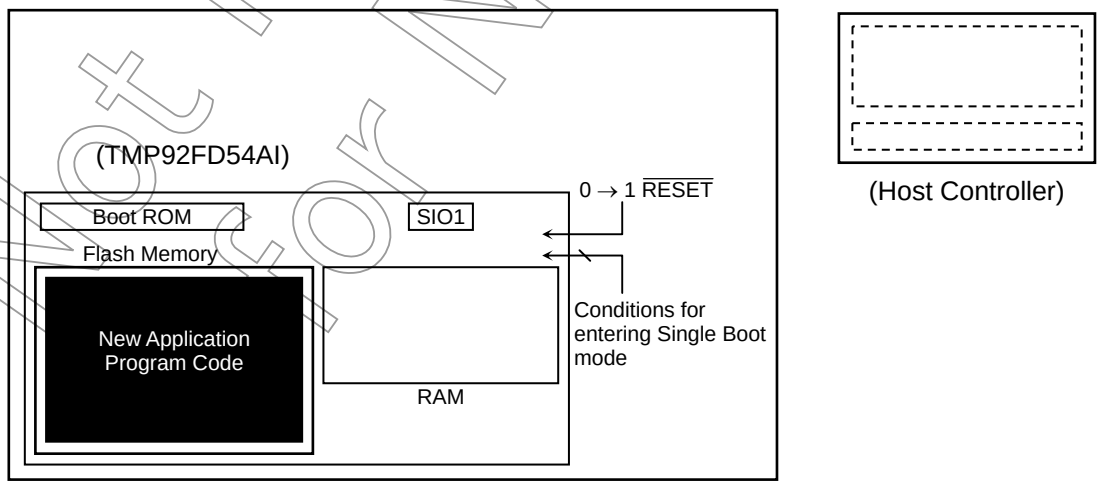


Figure 3.4.21 Executing the New Application Program

3.4.2.2 Connection Example in Single Boot Mode

In Single Boot mode, serial transfer is used to reprogram the flash memory while the TMP92FD54AI is installed on the board. In this mode, SIO1 of the TMP92FD54AI is connected to a host controller, which is to issue commands to the target board. Figure 3.4.22 shows an example of host-to-target connections.

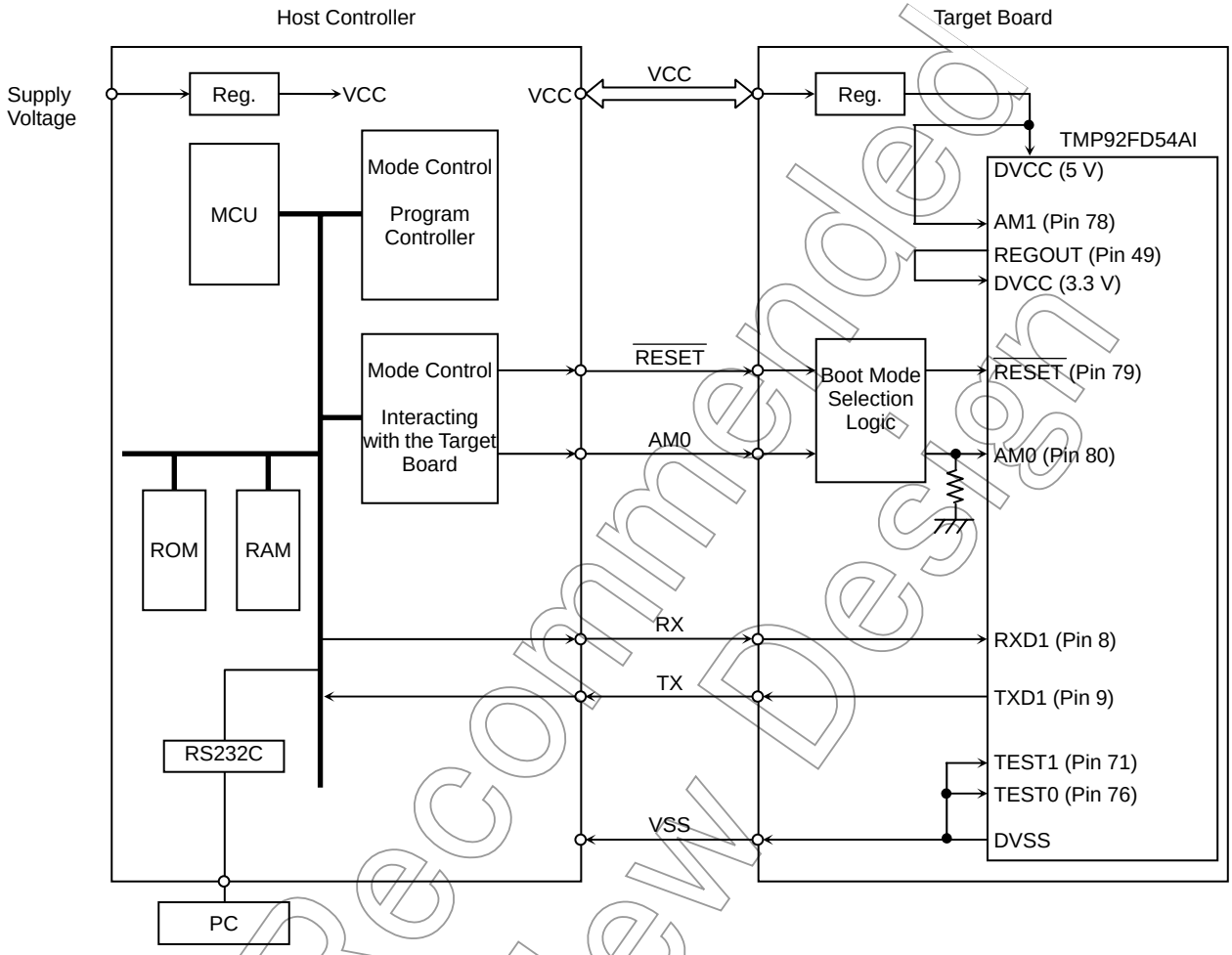


Figure 3.4.22 Example of a Connection between a Host Controller and a Target Board

(When the SIO1 is Configured for UART mode in Single Boot mode)

3.4.2.3 Mode Configuration

For on-board programming, boot the TMP92FD54AI in Single Boot mode as follows:

AM0, AM1 = H; TEST0, TEST1 = L; $\overline{\text{RESET}}$ = L → H

Set the $\overline{\text{RESET}}$ input at logic 0, while setting the AM0, AM1, TEST0 and TEST1 inputs at the logic values shown above. Then release $\overline{\text{RESET}}$ (high) so that the TMP92FD54AI reboots in Single Boot mode.

3.4.2.4 Memory Map

Figure 3.4.23 shows a comparison of the memory maps in Single-Chip and Single Boot modes. In Single Boot mode, the on-chip flash memory is mapped to the addresses 010000H through 08FFFFH; while the on-chip boot ROM (masked ROM) is mapped to the addresses FFF400H through FFFFFFH.

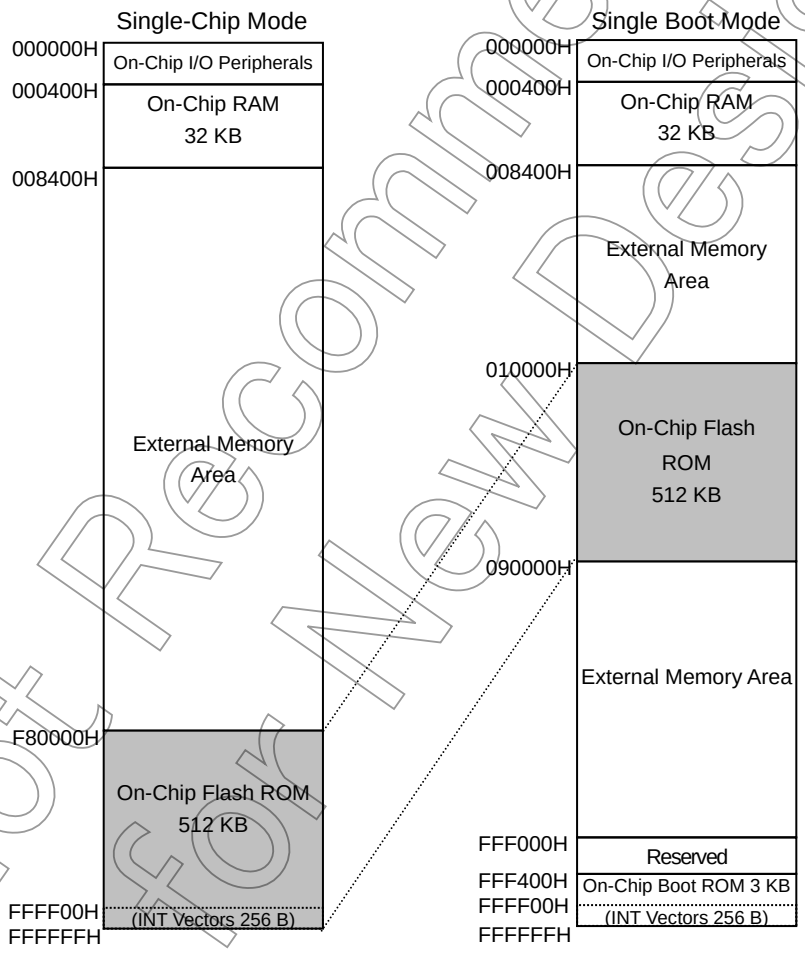


Figure 3.4.23 Memory Maps (for Single-Chip and Single Boot Modes)

3.4.2.5 Interface Specification

The SIO communication format in Single Boot mode is shown below. The serial transfer supports the UART (asynchronous) mode. To perform on-board programming, the host controller must also be configured to use the following communication format.

- UART mode

  Communication channel: SIO channel 1
  Serial transfer mode: UART (asynchronous) mode, full-duplex
  Data length: 8 bits
  Parity bits: None
  STOP bits: 1 bit
  Baud rate: Refer to Table 3.4.1

Table 3.4.1 Selective Baud Rates

Baud Rate (bps)	38400	19200	9600	4800	2400

Table 3.4.2 Pin Connections Required in Single Boot Mode

Power supply pins	DVCC3/5 (3.3/5.0 V)
	DVSS
Mode-setting pins	AM1, AM0, TEST1, TEST0
Reset pin	$\overline{\text{RESET}}$
Communications pins	TXD1
	RXD1

3.4.2.6 Data Transfer Formats

Operation commands and data transfer formats for each operation mode are shown in Table 3.4.3 through Table 3.4.7. At the same time, read the descriptions on the boot program commands described later.

Table 3.4.3 Operation Commands

Code	Command
10H	RAM Transfer
20H	Show Flash Memory Sum
30H	Show Product Information
40H	Auto Chip Erase & Unprotect

Table 3.4.4 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMP92FD54AI	Baud Rate	Data Transferred from the TMP92FD54AI to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK for the serial operation mode byte For UART Normal acknowledge 86H (The boot program aborts if the baud rate can not be set correctly.)
	3rd byte	Command code (10H)		—
	4th byte	—		ACK for the command code byte (Note 2) Normal acknowledge 10H Negative acknowledge x1H Communication error x8H
	5th byte thru 16th byte	Password sequence (12 bytes) (08FEF4H thru 08FEFFH)		—
	17th byte	Checksum value for bytes 5 - 16		
	18th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge 11H Communication error 18H
	19th byte	RAM storage start address (bits 31-24) (Note 3)		—
	20th byte	RAM storage start address (bits 23-16) (Note 3)		—
	21st byte	RAM storage start address (bits 15-8) (Note 3)		—
	22nd byte	RAM storage start address (bits 7-0) (Note 3)		—
	23rd byte	RAM storage byte count (bits 15-8) (Note 3)		—
	24th byte	RAM storage byte count (bits 7-0) (Note 3)		—
	25th byte	Checksum value for bytes 19-24 (Note 3)		—
	26th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge 11H Communication error 18H
	27th byte thru mth byte	RAM storage data		—
	(m + 1)th byte	Checksum value for bytes 27-m		—
	(m + 2)th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge 11H Communication error 18H
RAM	(m + 3)th byte	—		Jump to RAM storage start address

Note 1: For baud rate options, see Table 3.4.1.
Note 2: In case of any negative acknowledge the boot program returns to a state in which it waits for a command code (the 3rd byte).
Note 3: The 19th to 25th bytes must be within the RAM address range between 000400H and 006BFFH.

Table 3.4.5 Transfer Format for the Show Flash Memory Sum Command

	Byte	Data Transferred from the Controller to the TMP92FD54AI	Baud Rate	Data Transferred from the TMP92FD54AI to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK for the serial operation mode byte For UART Normal acknowledge 86H (The boot program aborts if the baud rate can not be set correctly.)
	3rd byte	Command code (20H)		—
	4th byte	—		ACK for the command code byte (Note 2) Normal acknowledge 20H Negative acknowledge x1H Communication error x8H
	5th byte	—		SUM (upper byte)
	6th byte	—		SUM (lower byte)
	7th byte	—		Checksum value for bytes 5 and 6
	8th byte	(Wait for the next command code.)		—

Note 1: For baud rate options, see Table 3.4.1.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (the 3rd byte).

Table 3.4.6 Transfer Format for the Show Product Information Command (1 of 2)

	Byte	Data Transferred from the Controller to the TMP92FD54AI	Baud Rate	Data Transferred from the TMP92FD54AI to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK for the serial operation mode byte For UART Normal acknowledge 86H (The boot program aborts if the baud rate can not be set correctly.)
	3rd byte	Command code (30H)		—
	4th byte	—		ACK for the command code byte (Note 2) Normal acknowledge 30H Negative acknowledge x1H Communication error x8H
	5th byte	—		Flash memory data (at address 08FEF0H)
	6th byte	—		Flash memory data (at address 08FEF1H)
	7th byte	—		Flash memory data (at address 08FEF2H)
	8th byte	—		Flash memory data (at address 08FEF3H)
	9th byte thru 20th byte	—		Product name (12-byte ASCII code) TMP92FD54AI_ from the 9th byte (where _ denotes a space)
	21st byte thru 24th byte	—		Password comparison start address (4 bytes) F4H, FEH, 08H and 00H from the 21st byte
	25th byte thru 28th byte	—		RAM start address (4 bytes) 00H, 04H, 00H and 00H from the 25th byte
	29th byte thru 32nd byte	—		Dummy data (4 bytes) FEH, 6BH, 00H and 00H from the 29th byte
	33rd byte thru 36th byte	—		RAM end address (4 bytes) FFH, 83H, 00H and 00H from the 33rd byte
	37th byte thru 40th byte	—		Dummy data (4 bytes) 00H, 00H, 00H and 00H from the 37th byte
	41st byte thru 44th byte	—		Dummy data (4 bytes) 00H, 00H, 00H and 00H from the 41st byte
	45th byte thru 46th byte	—		Protect status (2 bytes) From the 45th byte Protected 00H, 03H Not protected 00H, 01H
	47th byte thru 50th byte	—		Flash memory start address (4 bytes) 00H, 00H, 01H and 00H from the 47th byte
	51st byte thru 54th byte	—		Flash memory end address (4 bytes) FFH, FFH, 08H and 00H from the 51st byte
	55th byte thru 56th byte	—		Flash memory block count (2 bytes) 0AH and 00H from the 55th byte
	57th byte thru 60th byte	—		Start address of a group of the same-size flash blocks (4 bytes) 00H, 00H, 01H and 00H from the 57th byte

Table 3.4.7 Transfer Format for the Show Product Information Command (2 of 2)

	Byte	Data Transferred from the Controller to the TMP92FD54AI	Baud Rate	Data Transferred from the TMP92FD54AI to the Controller
Boot ROM	61st byte thru 64th byte	—	Desired baud rate (Note 1)	Size (in words) of the same-size flash blocks (4 bytes) 00H, 80H, 00H and 00H from the 61st byte
	65th byte	—		Number of flash blocks of the same-size flash blocks (1 byte) 06H
	66th byte thru 69th byte	—		Start address of a group of the same-size flash blocks (4 bytes) 00H, 00H, 07H and 00H from the 66th byte
	70th byte thru 73rd byte	—		Size (in words) of the same-size flash blocks (4 bytes) 00H, 70H, 00H and 00H from the 70th byte
	74th byte	—		Number of flash blocks of the same-size flash blocks (1 byte) 02H
	75th byte thru 78th byte	—		Start address of a group of the same-size flash blocks (4 bytes) 00H, C0H, 08H and 00H from the 75th byte
	79th byte thru 82nd byte	—		Size (in words) of the same-size flash blocks (4 bytes) 00H, 10H, 00H and 00H from the 79th byte
	83rd byte	—		Number of flash blocks of the same-size flash blocks (1 byte) 01H
	84th byte	—		Checksum value for bytes 5 to 83
	85th byte	(Wait for the next command code)		—

Note 1: For baud rate options, see Table 3.4.1.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (the 3rd byte).

Table 3.4.8 Transfer Format for the Auto Chip Erase & Unprotect Command

	Byte	Data Transferred from the Controller to the TMP92FD54AI	Baud Rate	Data Transferred from the TMP92FD54AI to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK for the serial operation mode byte For UART Normal acknowledge 86H (The boot program aborts if the baud rate can not be set correctly.)
	3rd byte	Command code (40H)		—
	4th byte	—		ACK for the command code byte (Note 2) Normal acknowledge 40H Negative acknowledge x1H Communication error x8H
	5th byte	—		ACK for the erase commands Normal acknowledge 4FH Negative acknowledge 4CH
	6th byte	—		ACK Normal acknowledge B1H Negative acknowledge B4H
	7th byte	(Wait for the next command code)		—

Note 1: For baud rate options, see Table 3.4.1.
Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (the 3rd byte).

3.4.2.7 Boot Program

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers the following commands:

1. RAM Transfer command

The RAM Transfer command stores program code (user program code) transferred from a host controller to the on-chip RAM. The CPU then jumps to the RAM storage start address and executes the program once the transfer is successfully completed. The user program of up to 26 Kbytes can be transferred. (The size is limited by the boot program for the protection of stack area and other areas. Once the RAM Transfer command is completed, the whole on-chip RAM is accessible.)

The RAM Transfer command can be used to download a flash program/erase routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The program/erase routine must utilize the flash memory command sequences described in Section 3.5, Flash Memory Program/Erase in On-Board Programming Modes.

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 and the boot program waits for the next command code.

2. Show Flash Memory SUM command

The Show Flash Memory Sum command adds the contents of the 512 Kbytes of the flash memory together and returns the sum result. The boot program does not provide a command to read out the contents of the whole area in the flash memory. Instead, the Show Flash Memory Sum command can be used for software revision management.

3. Show Product Information command

The Show Product Information command provides information like a product name and on-chip memory configuration. This command also reads out the contents of the given flash memory locations at addresses 08FEF0H to 08FEF3H. In addition to the Show Flash Memory Sum command, these locations can be used for software revision management.

4. Auto Chip Erase & Unprotect command

The Auto Chip Erase & Unprotect command erases all blocks in the flash memory. If some blocks are protected, executing this command resets the all block-protection setting and erasing all blocks in the flash memory.

Since this command also serves to restore the boot program in case that a user forgets the password, password matching is not performed.

3.4.2.8 RAM Transfer Command (See Table 3.4.4)

1. From Host Controller to Target Board

The 1st byte determines the operation mode and baud rate. During the first-byte interval, SC1MOD0<RXE> is cleared to disable the SIO1 reception. The boot program monitors the logic state of the RXD1 pin and calculates the baud rate.

- Communication in UART mode
  Send, from the controller to the target board, 86H in UART data format at the desired baud rate. If the serial operation mode is determined as UART, then the boot program checks if the SIO1 can be programmed to the baud rate at which the 1st byte was transferred. If the first byte is not identified as 86H, or if that baud rate is not possible, the boot program aborts, disabling any subsequent communications.

2. From Target Board to Host Controller

The 2nd byte is an acknowledge (ACK) response to the 1st byte, which specifies the serial operation mode. If the first byte is determined to specify UART mode, and also if the SIO1 is programmable at the desired baud rate, the boot program sends back 86H to the controller.

- Baud rate determination

  According to the baud rate value calculated based on the first byte, the TMP92FD54AI checks if the SIO1 is programmable at the desired baud rate. If it is determined as possible, the on-chip boot program sets the baud rate value in the BR1CR and BR1ADD registers and sends back 86H in UART data format after enabling the SIO1 reception by setting SC1MOD0<RXE>. If the SIO1 is not programmable at that baud rate, the boot program simply aborts without any error indication.

  The controller allows for a time-out period of five seconds after the transmission of a first byte is completed. If 86H is not received within the allotted time-out period, the controller should give up the communication.

3. From Host Controller to Target Board

The 3rd byte is an operation command data. The code for the RAM Transfer command is 10H, which is transmitted from the controller.

4. From Target Board to Host Controller

The 4th byte is an ACK response to the 3rd byte. Its upper four bits hold the same values as the upper four bits of the previously issued command.

Before sending back the ACK, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H (bit 3 = 1) and returns to the state in which it waits for a command (the 3rd byte) again.

If the 3rd byte is equal to any of the command codes listed in Table 3.4.3, the boot program echoes it back to the controller as a normal ACK. When the RAM Transfer command was received, the boot program echoes back a value of 10H and then branches to the RAM Transfer routine. Once this branch is taken, a password check is done. Password checking is detailed in Section 3.4.2.14, Password.

If the 3rd byte is not a valid command, the boot program sends back x1H (bit 1 =

1) as an invalid command error response and returns to the state in which it waits for a command (the 3rd byte) again.

5. From Host Controller to Target Board

The 5th to 16th bytes are a 12-byte password. The 5th byte is compared to the contents of address 08FEF4H in the flash memory; the 6th byte is compared to the contents of address 08FEF5H in the flash memory; likewise, the 16th byte is compared to the contents of address 08FEFFH in the flash memory.

6. From Host Controller to Target Board

The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add all these bytes (unsigned addition), drop the carries and take the two's complement of the lower 8 bits of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in detail in Section 3.4.2.16, Checksum Calculation.

7. From Target Board to Host Controller

The 18th byte is an ACK response to the 5th to 17th bytes (ACK for the checksum byte). Its upper four bits are the same as those of the previously issued command, 0001B.

First, the RAM Transfer routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 18H (bit 3 = 1) as a communication error response and returns to the state in which it waits for the next command (the 3rd byte) again.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 17th bytes (8-bit unsigned addition with the carry dropped) must result in 00H. If it is not 00H, the RAM Transfer routine sends back 11H (bit 0 = 1) as a checksum error response to the controller and returns to the state in which it waits for a command (the 3rd byte) again.

Finally, the RAM Transfer routine examines the result of the password check. The following case is treated as a password error. In this case, the RAM Transfer routine sends back 11H (bit 0 = 1) to the controller as a password error response and returns to the state in which it waits for a command (the 3rd byte) again.

- Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

When all the above checks have been successful, the RAM Transfer routine returns a normal ACK (10H) to the controller.

8. From Host Controller to Target Board

The 19th to 22nd bytes indicate the start address of the RAM region where subsequent data should be stored. The 19th byte corresponds to bits 31-24 of the address, the 20th byte corresponds to bits 23-16 of the address, the 21st byte corresponds to bit 15-8 and the 22nd byte corresponds to bits 7-0 of the address. The start address, transmitted to the target board, must be within the RAM address range 000400H-006BFFH.

9. From Host Controller to Target Board

The 23rd and 24th bytes indicate the number of bytes that will be transferred in

blocks from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15-8; while the 24th byte corresponds to bits 7-0. The byte count, transmitted to the target board, must be within the RAM address range 000400H-006BFFH.

10. From Host Controller to Target Board

    The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together (unsigned addition), drop the carries and take the two's complement of the lower 8 bits of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in detail in Section 3.4.2.16, Checksum Calculation.

11. From Target Board to Host Controller

    The 26th byte is an ACK response to the 19th to 25th bytes (ACK for the checksum byte). Its upper four bits are the same as those of the previously issued command, 0001B.

    First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there was a receive error, the RAM Transfer routine sends back 18H (bit 3 = 1) as a communication error response and returns to the state in which it waits for a command (the 3rd byte) again.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. The checksum value for the 19th to 25th bytes (8-bit unsigned addition with the carry dropped) must result in 00H. If it is not 00H, the RAM Transfer routine sends back 11H (bit 0 = 1) to the controller as a checksum error response, and returns to the state in which it waits for a command (the 3rd byte) again.

    When the above checks have been successful, the RAM Transfer routine returns a normal ACK (10H) to the controller.

12. From Host Controller to Target Board

    The 27th to mth bytes from the controller are stored in the TMP92FD54AI on-chip RAM. Storage begins at the address specified by the 19th to 22nd bytes and continues for the number of bytes specified by the 23rd and 24th bytes.

13. From Host Controller to Target Board

    The (m+1)th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together (8-bit unsigned addition with the carry dropped) and take the two's complement of the lower 8 bits of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in detail in Section 3.4.2.16, Checksum Calculation.

14. From Target Board to Host Controller

    The (m+2)th byte is an ACK response to the 27th to (m+1)th bytes (ACK for the checksum byte). Its upper four bits are the same as those of the previously issued command, 0001B.

    First, the RAM Transfer routine checks for a receive error in the 27th to (m+1)th bytes. If there is a receive error, the RAM Transfer routine sends back 18H (bit 3 = 1) as a communication error response, and returns to the state in which it waits for a command (the 3rd byte) again.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. The checksum value for the 27th to (m+1)th bytes (8-bit unsigned addition with the carry dropped) must result in 00H. If it is not 00H, the RAM Transfer routine sends back 11H (bit 0 = 1) to the controller as a checksum error response, and returns to the state in which it waits for a command (the 3rd byte) again.

When the above checks have been successful, the RAM Transfer routine returns a normal ACK (10H) to the controller.

15. From Target Board to Host Controller

If the (m+2)th byte is a normal ACK, a branch is made to the address specified by the 19th to 22nd bytes after sending a normal ACK (10H).

3.4.2.9 Show Flash Memory Sum Command (See Table 3.4.5)

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. From Host Controller to Target Board

The 3rd byte is a command code. The code for the Show Flash Memory Sum command is 20H.

3. From Target Board to Host Controller

The 4th byte is an ACK response to the 3rd byte. Its upper four bits hold the same values as the upper four bits of the previously issued command.

Before sending back the ACK, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H (bit 3 = 1) as a communication error response, and returns to the state in which it waits for a command (the 3rd byte) again.

If the 3rd byte is equal to any of the command codes listed in Table 3.4.3, the boot program echoes it back as the normal ACK to the controller. In this case, the boot program echoes back a value of 20H and then branches to the Show Flash Memory Sum routine.

If the 3rd byte is not a valid command, the boot program sends back x1H (bit 0 = 1) to the controller as an invalid command error response, and returns to the state in which it waits for a command (the 3rd byte) again.

4. From Target Board to Host Controller

The Show Flash Memory Sum routine adds all the bytes of the flash memory together. The 5th and 6th bytes indicate the upper and lower bytes of this total sum respectively. For details on sum calculation, see Section 3.4.2.15, Calculation of the Show Flash Memory Sum Command.

5. From Target Board to Host Controller

The 7th byte is a checksum value for the 5th and 6th bytes. To calculate the checksum value, add the 5th and 6th bytes (8-bit unsigned addition with the carry dropped), and take the two's complement of the lower 8 bits of the total sum. Transmit this checksum value from the controller to the target board.

The target board waits for the next command code after transmitting the 7th byte.

3.4.2.10 Show Product Information Command (See Table 3.4.6 and Table 3.4.7)

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. From Host Controller to Target Board

   The 3rd byte is a command code. The code for the Show Product Information command is 30H.

3. From Target Board to Host Controller

   The 4th byte is an ACK response to the 3rd byte. Its upper four bits hold the same values as the upper four bits of the previously issued command.

   Before sending back the ACK, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H (bit 3 = 1) as a communication error response, and returns to the state in which it waits for a command (the 3rd byte) again.

   If the 3rd byte is equal to any of the command codes listed in Table 3.4.3, the boot program echoes it back to the controller as a normal ACK. In this case, the boot program echoes back a value of 30H and then branches to the Show Product Information routine.

   If the 3rd byte is not a valid command, the boot program sends back x1H (bit 0 = 1) to the controller as invalid command error response, and returns to the state in which it waits for a command (the 3rd byte) again.

4. From Target Board to Host Controller

   The 5th to 8th bytes are the data read from the addresses 08FEF0H to 08FEF3H in the flash memory. Software version management is possible by storing a software ID in these locations.

5. From Target Board to Host Controller

   The 9th to 20th bytes indicate the product name, which is TMP92FD54AI_ in ASCII code (where _ denotes a space).

6. From Target Board to Host Controller

   The 21st to 24th bytes indicate the password comparison start address, i.e., F4H, FEH, 08H, 00H.

7. From Target Board to Host Controller

   The 25th to 28th bytes indicate the start address of the on-chip RAM, i.e., 00H, 04H, 00H, 00H.

8. From Target Board to Host Controller

   The 29th to 32nd bytes are the end address of the on-chip RAM (in the user area), i.e., FFH, 6BH, 00H, 00H.

9. From Target Board to Host Controller

   The 33rd to 36th bytes indicate the end address of the on-chip RAM, i.e., FFH, 83H, 00H, 00H.

10. From Target Board to Host Controller

The 37th to 44th bytes are dummy data.

11. From Target Board to Host Controller

The 45th and 46th bytes indicate protect status of blocks in the flash memory. If any of Blocks 0 to 9 is protected, the 45th and 46th bytes contain 00H and 01H respectively. If all blocks are unprotected, they contain 00H and 03H respectively.

12. From Target Board to Host Controller

The 47th to 50th bytes indicate the start address of the on-chip flash memory, i.e., 00H, 00H, 01H, 00H.

13. From Target Board to Host Controller

The 51st to 54th bytes indicate the end address of the on-chip flash memory, i.e., FFH, FFH, 08H, 00H.

14. From Target Board to Host Controller

The 55th and 56th bytes indicate the number of flash memory blocks available, i.e., 0AH, 00H.

15. From Target Board to Host Controller

The 57th to 83rd bytes contain information about the flash blocks. Flash blocks of the same size are treated as a group. Information about the flash blocks indicate the start address of a group, the size of the blocks in that group (in words) and the number of the blocks in that group.

The 57th to 65th bytes are the information about the 64-Kbyte blocks (Block 0 to Block 5); the 66th to 74th bytes are the information about the 56-Kbyte blocks (Blocks 6 and 7); the 75th to 83rd bytes are the information about the 8-Kbyte blocks (Blocks 8 and 9). For the values of the transmitted bytes, refer to Table 3.4.6 and Table 3.4.7.

16. From Target Board to Host Controller

The 84th byte is a checksum value for the 5th to 83rd bytes. The checksum value is calculated by adding all these bytes together (8-bit unsigned addition with the carry dropped) and taking the two's complement of the lower 8 bits of the total sum. The checksum value is transmitted from the controller to the target board.

The target board waits for the next command code after transmitting the 84th byte.

3.4.2.11 Auto Chip Erase & Unprotect Command (Refer to Table 3.4.8)

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. From Host Controller to Target Board

   The 3rd byte indicates a command code. The code for the Auto Chip Erase & Unprotect command is 40H.

3. From Target Board to Host Controller

   The 4th byte is an ACK response to the 3rd byte. Its upper four bits hold the same values as the upper four bits of the previously issued command.

   Before sending back the ACK, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H (bit 3 = 1) as a communication error response, and returns to the state in which it waits for a command (the 3rd byte) again.

   If the 3rd byte is equal to any of the command codes listed in Table 3.4.3, the boot program echoes it back to the controller as a normal ACK. In this case, the boot program echoes back a value of 40H and then branches to the Auto Chip Erase & Unprotect routine.

   If the 3rd byte is not a valid command, the boot program sends back x1H (bit 0 = 1) to the controller as an invalid command error response, and returns to the state in which it waits for a command (the 3rd byte) again.

4. From Target Board to Host Controller

   The 5th byte indicates whether the Auto Chip Erase processing is properly completed. If it is completed, the end code (4FH) is sent back; while the error code (4CH) is sent back for processing error.

5. From Target Board to Host Controller

   The 6th byte is an ACK response. If a command is completed, the end code (B1H) is sent back as a normal ACK. If an error occurs, the error code (B4H) is sent back as an erase error response.

The target board waits for the next command code after transmitting the 6th byte.

3.4.2.12 Acknowledge (ACK) Responses

The boot program represents processing states with specific codes. Table 3.4.9 to Table 3.4.12 show the values of possible ACK responses to the received data. The upper four bits of the ACK are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and Bit 2 are always 0.

Table 3.4.9 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
86H	The SIO can be configured to operate in UART mode. (Note 1)

Note 1: If the SIO is not programmable at the baud rate, the program simply aborts with no error indication.

Table 3.4.10 ACK Response to the Command Byte

Return Value	Meaning
x8H (Note 2)	A receive error occurred while getting a command code.
x1H (Note 2)	An undefined command code was received.
10H	The RAM transfer command was received.
20H	The Show Flash Memory Sum command was received.
30H	The Show Product Information command was received.
40H	The Auto Chip Erase & Unprotect command was received.

Note 2: The upper four bits of the ACK are the same as those of the previous command code.

Table 3.4.11 ACK Response to the Checksum Byte

Return Value	Meaning
18H	A receive error occurred.
11H	A checksum error or password error occurred.
10H	The checksum was correct.

Table 3.4.12 ACK Response to the Auto Chip Erase & Unprotect Byte

Return Value	Meaning
4FH, B1H	An erase operation was properly completed.
4CH, B4H	An erase error occurred.

3.4.2.13 Automatic Baud Rate Programming for UART Mode

The controller must first send a value of 86H at a desired baud rate to the target board in UART data format. The waveform of the first byte, 86H, is shown in Figure 3.4.24.

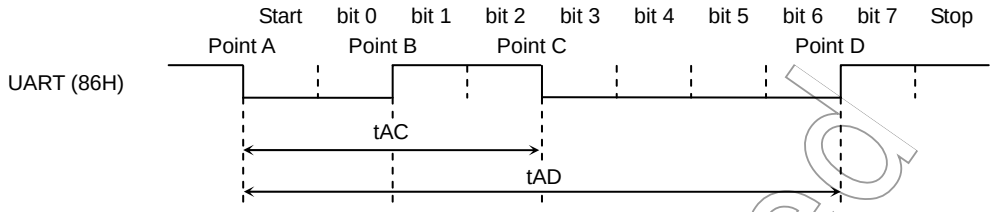


Figure 3.4.24 Serial Operation Mode Byte, 86H

After $\overline{\text{RESET}}$ is released, the boot program does not receive the first serial byte (86H) from the controller as a serial receive data. The boot program monitors the logic states of the RXD1 pin and captures timer counts each time a logic transition of the pin occurs.

The boot program calculates the intervals of tAB, tAC and tAD. Then, the serial operation mode is determined as UART mode. At the same time, the boot program determines whether the auto programming of baud rate is possible or not.

If the controller fails to receive an echo-back (86H), or if the boot program determines that the SIO is not programmable at the baud rate, the controller should give up further communication.

3.4.2.14 Password

The RAM Transfer command (10H) causes the boot program to perform a password check. Following an echo-back of the command code, the boot program checks the contents of the 12-byte password area (8FEF4H to 8FEFFH) within the flash memory.

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. Table 3.4.13 shows how they are compared byte-by-byte. All of the 12 bytes must match to pass the password check. Otherwise, a password error occurs, which causes the boot program to return an error acknowledge at the 18th byte in response to the checksum byte (the 17th byte).

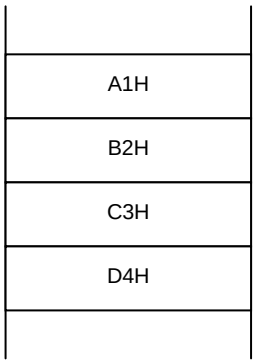
Table 3.4.13 Relationship between Received Bytes and Flash Memory Locations

Received Byte	Compared Flash Memory Data
5th byte	Address 08FEF4
6th byte	Address 08FEF5
7th byte	Address 08FEF6
8th byte	Address 08FEF7
9th byte	Address 08FEF8
10th byte	Address 08FEF9
11th byte	Address 08FEFA
12th byte	Address 08FEFB
13th byte	Address 08FEFC
14th byte	Address 08FEFD
15th byte	Address 08FEFE
16th byte	Address 08FEFF

3.4.2.15 Calculation of the Show Flash Memory Sum Command

The Show Flash Memory Sum command adds all 512 Kbytes of the flash memory together by performing an 8-bit unsigned addition, and provides the total sum as a word quantity. The sum is sent to the controller with the upper eight bits first, followed by the lower eight bits. The sum returned in response to the Show Flash Memory Sum command is calculated in this way.

Example:



For the interest of simplicity, assume the depth of the flash memory is four locations as shown in the figure on the left. Then the sum of the four bytes is calculated as:

A1H + B2H + C3H + D4H = 02EAH

Hence, 02H is first sent to the controller, followed by EAH.

3.4.2.16 Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together (8-bit unsigned addition with the carry dropped), and taking the two's complement of the lower 8 bits. The Show Flash Memory Sum command and the Show Product Information command perform the checksum calculation. The controller must perform the same checksum operation when transmitting checksum bytes.

Example:

Assume the Show Flash Memory Sum command provides the upper and lower bytes of the sum as E5H and F6H. To calculate the checksum for a series of E5H and F6H:

Add the bytes together (using an 8-bit unsigned addition).

E5H + F6H = 1DBH

Take the two's complement of the sum of the lower 8 bits, and that is the checksum byte. Hence, 25H is sent to the controller.

0 – DBH = 25H

3.4.2.17 General Boot Program Flowchart

Figure 3.4.25 shows an overall flowchart of the boot program.

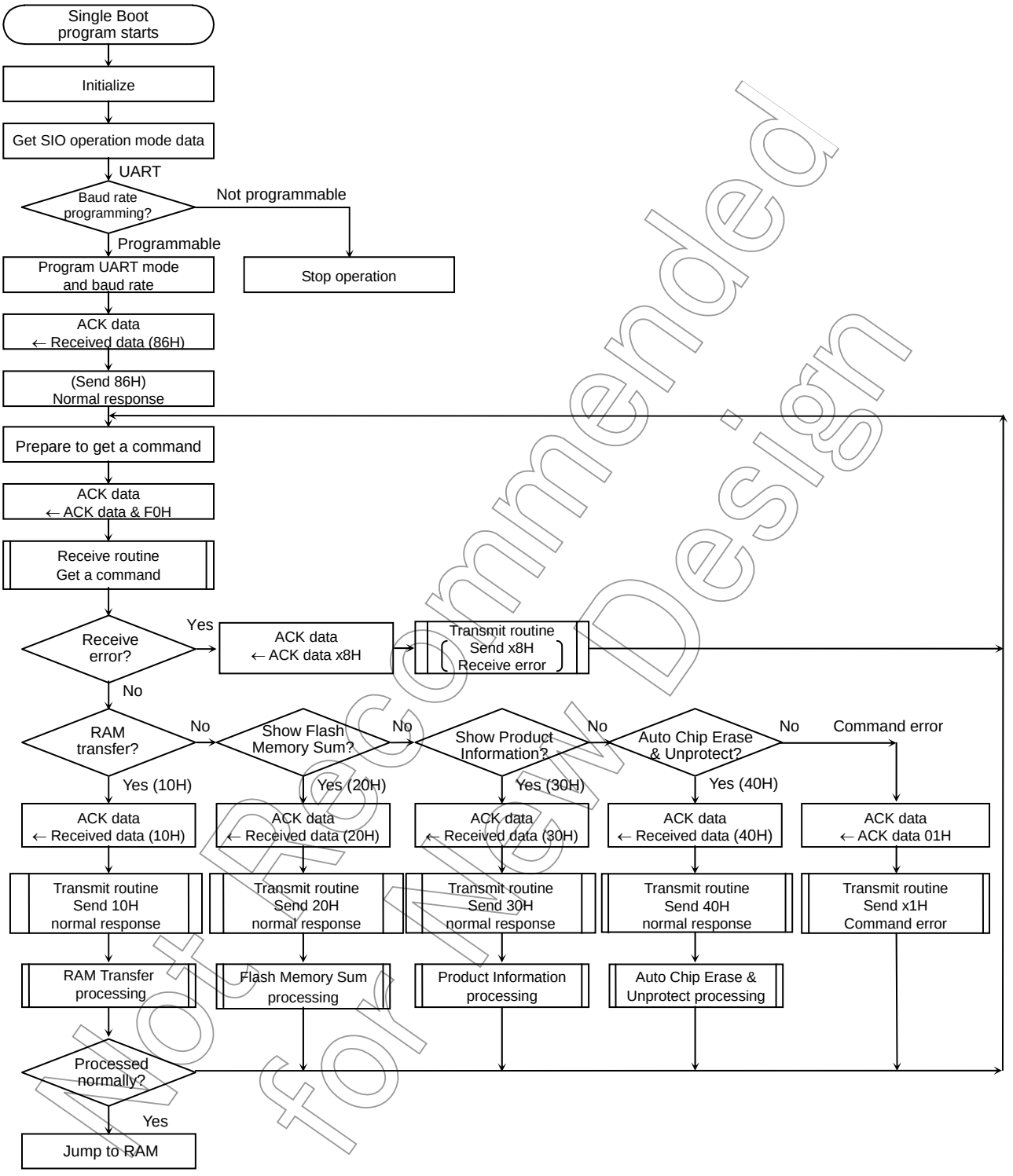


Figure 3.4.25 Overall Boot Program Flow

3.5 Flash Memory Program/Erase in On-Board Programming Modes

The TMP92FD54AI flash memory is command set compatible with the JEDEC EEPROM standard. The flash memory can be programmed and erased by the CPU executing software commands.

Because the flash memory cannot be read while it is being programmed or erased, the program/erase routine must be stored in the on-chip RAM or an external memory device.

Note: It is the user's responsibility to prepare a program/erase routine.

3.5.1 Key Features

The TMP92FD54AI flash memory commands are in principle compatible with the standard JEDEC commands with a few exceptions. The system can issue a command sequence to the flash memory by using CPU instructions such as LD. According to the written command sequence, the flash memory initiates the embedded program or erase algorithm automatically.

Table 3.5.1 Flash Memory Features

Feature	Description
Auto Program	Programs and verifies the specified addresses in longword (32-bit) quantities automatically.
Auto Chip Erase	Erases and verifies the entire memory array automatically.
Auto Block Erase	Erases and verifies all memory locations in the selected block automatically.
Hardware sequence flag	Provides several status bits such as the Data Polling bit, which can be used to determine whether a program or erase operation is complete or in progress.
Block Protect	Disables both program and erase operations in any block.
Auto Chip Erase & Unprotect	Erases the entire memory array and unprotects all blocks automatically.

Due to the on-chip CPU interface, the TMP92FD54AI uses addresses different from those of the JEDEC standard flash command sequences. Unless otherwise noted, programming is done in 32-bit quantities; thus the 32-bit load instruction should be used. The byte (8-bit) load instruction can be used to issue commands to the flash memory.

3.5.2 Block Architecture

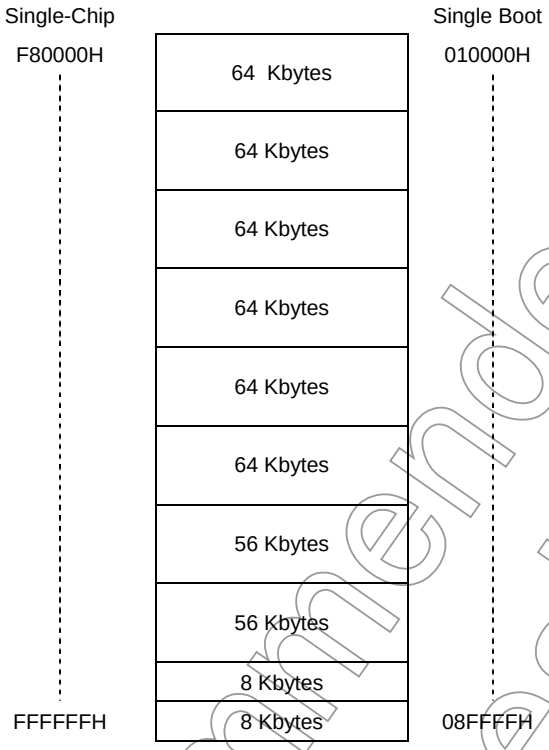


Figure 3.5.1 Flash Memory Block Architecture

3.5.3 Internal CPU-to-Flash Interface

Figure 3.5.2 illustrates the internal interface between the CPU and the flash memory of the TMP92FD54AI. The diagram does not show the actual logic network; instead it is only a conceptual depiction of the CPU-to-flash interface.

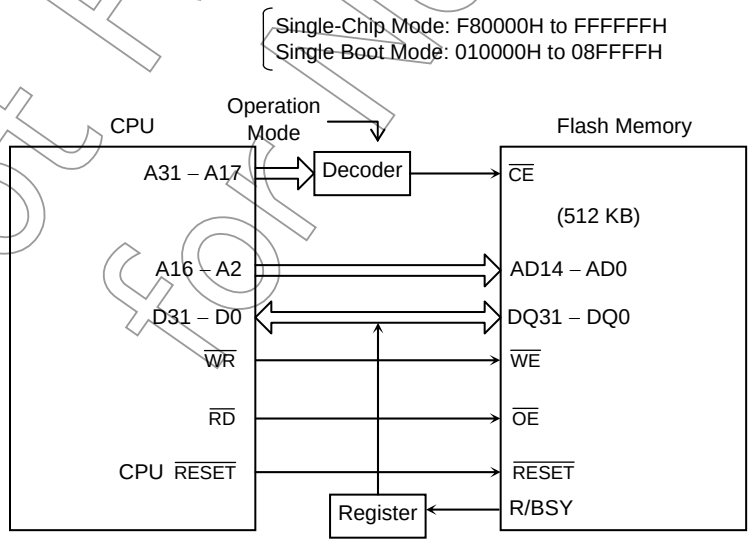


Figure 3.5.2 Internal CPU-to-Flash Interface

3.5.4 Read Mode and Automatic Operation Mode

The flash memory of the TMP92FD54AI has the following two operation modes:

- Read mode: Array data is read.
  (The flash memory cannot be reprogrammed in this mode.)
- Automatic Operation mode: The flash array is automatically programmed or erased.
  (Array data cannot be read in this mode.)

The flash memory enters Automatic Operation mode when a valid command is executed in Read mode.

1) Read

For data read operations, the flash memory enters Read mode upon the following conditions:

a) The automatic operation is successfully completed.
b) Hardware reset (Include the CPU internal reset)
c) Software reset (Read/Reset command)

2) Writing Commands

The TMP92FD54AI flash memory is command set compatible with the JEDEC EEPROM standard. Write operations to the internal command register is performed by executing command sequences to the flash memory. The flash memory executes command sequences by latching the written addresses and data into the command register. (See Table 3.5.3 and Table 3.5.4.)

Commands are written via DQ0-DQ7 except the fourth (read) cycle in the Read/Reset command sequence, the fourth (write) cycle in the Auto Program command sequence and the fourth (write) cycle in the Verify Block Protect command sequence. Thus, commands can be provided byte by byte.

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.

3) Reset

Resetting the flash memory can put it back into Read mode. The flash memory should be reset after executing the Verify Block Protect command, or if an automatic operation of the flash memory terminated abnormally.

- Read/Reset command (software reset)

The Read/Reset command must be issued to initialize the internal circuitry of the flash memory and put the flash memory back in Read mode. Since the flash memory disables command sequence receptions while an automatic operation is in progress, the flash memory cannot be reset with the Read/Reset command during that period.

- Hardware Reset ($\overline{\text{RESET}}$ pin input)

Asserting the $\overline{\text{RESET}}$ pin initializes the internal circuitry of the flash memory and puts it back to Read mode by hardware.

As shown in Figure 3.5.2, the flash memory has a reset pin, which is connected to the reset signal of the CPU. When the system drives the $\overline{\text{RESET}}$ pin to V_{IL} or when certain events such as a watchdog timer time-out causes a CPU reset, a hardware reset is performed on the flash memory.

A hardware reset operation immediately terminates any operation in progress and the flash memory is reset to Read mode. If a hardware reset is performed in the middle of an automatic operation, the flash memory data might be corrupted due to an abort of any ongoing operation, such as program and erase operations. Thus, after performing a hardware reset, it should be checked whether there is any change in the entire memory array and their protect status.

4) Auto Program Command

The Auto Program operation cannot overwrite programmed memory locations. 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 auto erase operation can change a 0 back to a 1.

In on-board programming modes, the Auto Program command programs the desired addresses in longword (32-bit) quantities. Thus, the program address must be a multiple of four.

The Auto Program command requires four bus cycles; the program address and data are written in the fourth cycle, upon completion of which the program operation will commence.

The Auto Program command executes a sequence of events to program the desired bits of the addressed memory location and verify that the desired bits are sufficiently programmed. The block protection feature disables programming operations in any block. If an attempt is made to program a protected block, the Auto Program command does nothing; the flash memory returns to Read mode in approximately 3 μs after the completion of the fourth bus cycle of the command sequence.

Any commands written during the programming operation are ignored. The system can determine the status of the programming operations by using write status flags and the FLSR<R/BSY> bit.

When the embedded Auto Program algorithm is complete, the flash memory returns to Read mode. If any failure occurs during the programming operation, the flash memory remains locked in Automatic Operation mode. The system can determine this status by using write status flags and the FLSR<R/BSY> bit.

To put the flash memory back in Read mode from the locked state, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of a programming failure, it is recommended to replace the chip or discontinue the use of the flash block including this failing address.

5) Auto Chip Erase Command

The Auto Chip Erase operation will commence after completion of the sixth bus write cycle. The embedded Auto Chip Erase algorithm automatically preprograms the entire memory for all-0 data pattern prior to the erase; then it automatically erases and verifies the entire memory. The Auto Chip Erase algorithm erases the unprotected blocks and ignores the protected blocks. If all the blocks are protected, the Auto Chip Erase command does nothing; the flash memory returns to Read mode after the completion of the sixth bus write cycle of the command sequence.

Any commands written during the chip erase operation are ignored. The system can determine the status of the chip erase operation by using write status flags or the FLSR<R/BSY> bit.

When the embedded Auto Chip Erase algorithm is complete, the flash memory returns to Read mode. 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 and the FLSR<R/BSY> bit.

To put the flash memory back in Read mode from the locked state, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In this case, however, the failing block cannot be identified by means of the Auto Chip Erase command. It is recommended to replace the chip or discontinue the use of the failing flash block by identifying the failing block by running the Auto Block Erase command.

6) Auto Block Erase Command

The Auto Block Erase operation will commence after completion of the sixth bus write cycle of the command sequence.

The embedded Auto Block Erase algorithm automatically preprograms the selected block for all-0 data pattern, and then erases and verifies that block. The block protection feature disables erase operations in the block. If the selected block is protected, the Auto Block Erase algorithm does nothing; flash memory returns to Read mode after the completion of the sixth bus write cycle of the command sequence.

Any commands written during the block erase operation are ignored. The system can determine the status of the erase operation by using write status flags and the FLSR<R/BSY> bit.

When the embedded Auto Block Erase algorithm is complete, the flash memory automatically returns to Read mode. If any failure occurs during the erase operation, the flash memory remains locked in Automatic Operation mode. The system can determine this status by using write status flags or the FLSR<R/BSY> bit.

To put the flash memory back in Read mode from the locked state, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In case of an erase failure, it is recommended to replace the chip or discontinue the use of the failing flash block.

Table 3.5.2 Block Erase Addresses

Block	Address Range		Size
	User Boot Mode	Single Boot Mode	
BA0	F80000H - F8FFFFH	010000H - 01FFFFH	64 Kbytes
BA1	F90000H - F9FFFFH	020000H - 02FFFFH	64 Kbytes
BA2	FA0000H - FAFFFFH	030000H - 03FFFFH	64 Kbytes
BA3	FB0000H - FBFFFFH	040000H - 04FFFFH	64 Kbytes
BA4	FC0000H - FCFFFFH	050000H - 05FFFFH	64 Kbytes
BA5	FD0000H - FDFFFFH	060000H - 06FFFFH	64 Kbytes
BA6	FE0000H - FEDFFFH	070000H - 07DFFFH	56 Kbytes
BA7	FEE000H - FFBFFFH	07E000H - 08BFFFH	56 Kbytes
BA8	FFC000H - FFDFFFH	08C000H - 08DFFFH	8 Kbytes
BA9	FFE000H - FFFFFFH	08E000H - 08FFFFH	8 Kbytes

The address of the block to be erased can be any of the addresses within that block with A0 = 0 and A1 = 0. For example, to select BA0 in User Boot mode, provide any address with A0 = 0 and A1 = 0 in the range between F80000H and F8FFFFH.

7) Auto Block Protect Command

The auto block protection feature disables program and erase operations in any block independently from other blocks. The effects of the program and erase commands on the protected/unprotected blocks are summarized below.

Table 3.5.3 Effects of the Program and Erase Commands on the Protected Blocks

Command and protected/unprotected block status	Operation
Program command on a protected block	No programming operation is performed, and the flash memory automatically returns to Read mode.
Block erase command on a protected block	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip erase command when all the blocks are protected	No erase operation is performed, and the flash memory automatically returns to Read mode.
Chip erase command when any blocks are protected	Only the unprotected blocks are erased. Upon completion, the flash memory automatically returns to Read mode.

The Auto Block Protect command requires six bus write cycles; the block address and command data are written in the sixth cycle, upon completion of which the block protect operation will commence.

The embedded Auto Block Protect algorithm automatically performs block protect operations and verifies that protected blocks.

Any commands written during the Auto Block Protect algorithm are ignored. The system can determine the state of the Auto Block Protect operation by using write status flags or the FLSR<R/BSY> bit.

If any failure occurs during the protect operation, the flash memory remains locked in Automatic Operation mode. The system can determine this status by using write status flags or the FLSR<R/BSY> bit.

8) Verify Block Protect Command

The Verify Block Protect command is used to verify the protect status of a block. The address of the block to be verified is given in the fourth read cycle. Any address within the block range will suffice, provided A0 = A1 = A2 = A3 = 0, A4 = 1 and A6 = 0. A longword (32-bit) read should be performed on that address. If the selected block is protected, a value

of 0000_0001H is returned. If the selected block is not protected, a value of 0000_0000H is returned.

Additional blocks may be verified by repeating the fourth bus read cycle of the command sequence. Longword reads should be performed on addresses within the range of desired blocks. The Verify Block Protect command does not return the flash memory to Read mode. Either the Read/Reset command or a hardware reset is required to reset the flash memory to Read mode or to write the next command sequence.

9) Hardware sequence Flags (Refer to Table 3.5.6)

The flash memory provides several flag bits to determine the status of an automatic operation. These status bits can be read during an automatic operation using the same timing as for Read mode. The flash memory automatically returns to Read mode when an automatic operation completes.

During the automatic program operation, the system must provide the program address (with A0 = 0 and A1 = 0) to read valid status information.

- DQ7 (Data Polling)

  The Data Polling bit, DQ7, indicates the status of the automatic operation.

  Data Polling is valid after the final bus write cycle of an automatic command sequence. When the embedded Auto Program algorithm is in progress, an attempt to read the flash memory will produce the complement (inverted) data last written to DQ7. Upon completion of the embedded Auto Program algorithm, an attempt to read will produce the true (non-inverted) data. Therefore, the system can use DQ7 to determine whether the embedded Auto Program algorithm is in progress or complete. When the embedded Auto 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 Auto Erase algorithm, the flash memory will produce a 1 at the DQ7 output. If there is a failure during an automatic operation, DQ7 continues to output the same value. Thus, DQ7 must always be polled in conjunction with the Exceeded Timing Limits (DQ5) flag. (See Figure 3.5.9).

  The flash memory disables address latching when an automatic 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.

  The system cannot properly determine the status of the Auto Block Protect operation by using the DQ7 bit. The Toggle bit and the FLSR<R/BSY> bit must be checked to determine the correct status.

- DQ6 (Toggle Bit)

  Basically same as the Data Polling bit, the Toggle bit, DQ6, indicates the status of the automatic operation. Toggle Bit is valid after the final bus write cycle of an automatic command sequence. Every time the flash memory is read while the automatic program algorithm is in progress, the DQ6 output toggles between 1 and 0. If there is a failure during an automatic operation, DQ6 continues to output the same value. Thus, DQ6 must be used in conjunction with the Exceeded Timing Limits (DQ5) flag. (See Figure 3.5.10)

- 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 automatic operation is not successfully completed, including the possibility of flash memory failure. A bit in the flash memory cells can be programmed from a 1 to a 0, but not vice versa. Thus, if the system tries to program a 1 to a location that was previously programmed to a 0, the operation cannot be completed within the specified timing limits and DQ5 will indicate a 1. Note that this is not a device failure condition but the flash memory was used incorrectly.

- DQ3 (Block Erase Timer)

  When the erase operation begins, DQ3 switches from a 0 to a 1. When the other operations are in progress (such as the Auto Program operation), the flash memory produces a 0 at the DQ3 output.

  If there is a failure during the Auto Erase operation, DQ3 remains 1.

Table 3.5.4 Hardware sequence flags

Status		D7 (DQ7)	D6 (DQ6)	D5 (DQ5)	D3 (DQ3)
Automatic operation in progress	Auto Program	$\overline{DQ7}$	Toggles	0	0
	Auto Erase	0	Toggles	0	1
	Auto Block Protect	Note 2	Toggles	0	0
	Auto Chip Erase & Unprotect	0	Toggles	0	1
Time-out in automatic operation (failure)	Auto Program	$\overline{DQ7}$	Toggles	1	0
	Auto Erase	0	Toggles	1	1
	Auto Block Protect	Note 2	Toggles	1	0
	Auto Chip Erase & Unprotect	0	Toggles	1	1
Operation complete	Read	Cell data	Cell data	Cell data	Cell data

Note 1: D31-D8, D4 and D2-D0 are don't-cares.
Note 2: The automatic operation status cannot be determined with the Data Polling bit, DQ7.

10) Status Register

This is an 8-bit register that indicates the Ready/Busy status of an automatic operation algorithm.

FLSR (16EH)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	R/BSY	-	-
Read/Write	R/W	R/W	R/W	R/W		R		
After Reset	0	0	0	0	-	1	-	-
Function	Must be written as 0.	Must be written as 0.	Must be written as 0.	Must be written as 0.		Ready/Busy 0: Automatic operation algorithm is in progress. 1: Automatic operation algorithm is complete.		

Figure 3.5.3 Flash Status Register

- Bit 2: R/BSY Flag Bit

The flash memory provides the R/BSY flag bit. The CPU monitors this bit to determine whether an automatic algorithm is in progress or complete. The R/BSY bit is cleared to a 0 when the flash memory is actively erasing or programming. The R/BSY bit is set to a 1 when an automatic operation has completed and the flash memory is ready to accept the next command. If any failure occurs during the program or erase operation, this bit remains cleared. A hardware reset sets this bit.

The R/BSY bit is cleared upon completion of the final bus write cycle of an automatic operation command. Any command is ignored while the R/BSY bit is cleared.

11) Flash Security Enable Register

This is an 8-bit register to enable or disable the Auto Chip Erase & Unprotect operation.

FSWE (16BH)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	0	0	0	0	0	0	0
Function	C9H: Auto Chip Erase & Unprotect operation is enabled. Other: Auto Chip Erase & Unprotect operation is disabled.							

Figure 3.5.4 Flash Security Enable Register

The FSWE register can only be written by executing the Auto Program algorithm out of the on-chip RAM or an external memory device.

12) Auto Chip Erase & Unprotect Command

The FSWE register must be set to C9H to start the Auto Chip Erase & Unprotect operation after the completion of the sixth bus write cycle of the command sequence. The Auto Chip Erase & Unprotect algorithm automatically unprotects all the flash blocks, erases the entire flash array, then verifies if the entire flash memory is cleared. Any commands written during the chip erase operation are ignored.

If the Auto Chip Erase & Unprotect operation is successfully completed, the flash memory automatically returns to Read mode. The FSWE register must be cleared (or programmed to any value other than C9H) if any on-board operation is subsequently required. If any failure occurs during the Auto Chip Erase & Unprotect operation, the flash memory remains locked in Automatic Operation mode. The system can determine this status by using hardware sequence flags or the FLSR<R/BSY> bit..

To put the flash memory back in Read mode, use the Read/Reset command to reset the flash memory or a hardware reset to reset the whole chip. In this case, however, there is no way to identify a failing flash block. It is recommended to replace the chip or discontinue the use of the failing flash block by identifying the failing block by means of the Block Erase command.

13) Command Sequences

The command sequences and addresses for on-board programming are shown in the following tables.

Table 3.5.5 On-Board Programming Mode Command Sequences

Command Sequence	Cycles Required	1st Cycle (Write)		2nd Cycle (Write)		3rd Cycle (Write)		4th Cycle (Read/Write)		5th Cycle (Read/Write)		6th Cycle (Read/Write)	
		Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
Read/Reset	1	xXX00H	F0H										
Read/Reset	3	xAAA8H	AAH	x5554H	55H	xAAA8H	F0H	RA	RD				
Auto Program	4	xAAA8H	AAH	x5554H	55H	xAAA8H	A0H	PA	PD				
Auto Chip Erase	6	xAAA8H	AAH	x5554H	55H	xAAA8H	80H	xAAA8H	AAH	x5554H	55H	xAAA8H	10H
Auto Block Erase	6	xAAA8H	AAH	x5554H	55H	xAAA8H	80H	xAAA8H	AAH	x5554H	55H	BA	30H
Block Protect	6	xAAA8H	AAH	x5554H	55H	xAAA8H	9AH	xAAA8H	AAH	x5554H	55H	BA	70H
Verify Block Protect	4	xAAA8H	AAH	x5554H	55H	xAAA8H	90H	BPA	BD				
Auto Chip Erase& Unprotect (Note 1)	6	xAAA8H	AAH	x5554H	55H	xAAA8H	80H	xAAA8H	AAH	x5554H	55H	xAAA8H	10H

Note 1: The FSWE register must be programmed to C9H before executing the Auto Chip Erase & Unprotect command sequence.

Note 2: There must be an interval of at least two instructions between each bus cycle.

Table 3.5.6 Addresses Provided by the CPU

Command Address	CPU Addresses: A23-A0																
Address	A23-A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
xXXXXH	Flash memory block	X				X				X				X			
x0000H		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
xAAA8H		1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
X5554H		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0

- AAH, 55H, F0H, A0H, 10H, 30H, 70H, 80H, 90H, 9AH — Command data. Write command data as a byte quantity.
- RA: Read address
  RD: Read data
- PA: Program address
  PD: Program data — The address must be a multiple of four. Write data on a word-by-word basis.
- BA: Block address (BA0-BA9) — Refer to Table 3.5.2, Block Erase Addresses.
- BPA: Verify Block Protect Address — The address of the block to be verified can be any of the addresses within the block, with A6 = 0, A4 = 1, A3 = 0 and A2 = A1 = A0 = 0.
- BD: Block Protect Data — If a block is protected, a value of 0000_0001H will be returned. If a block is not protected, a value of 0000_0000H will be returned.

14) Automatic operation algorithms

Figure 3.5.5 shows a flowchart of the Auto Program operation. For details on the step "Determine the Status of an Automatic Operation," see Figure 3.5.9, Data Polling (DQ7) Algorithm, Figure 3.5.10, Toggle Bit (DQ6) Algorithm, and Figure 3.5.11, R/BSY Flag Algorithm.

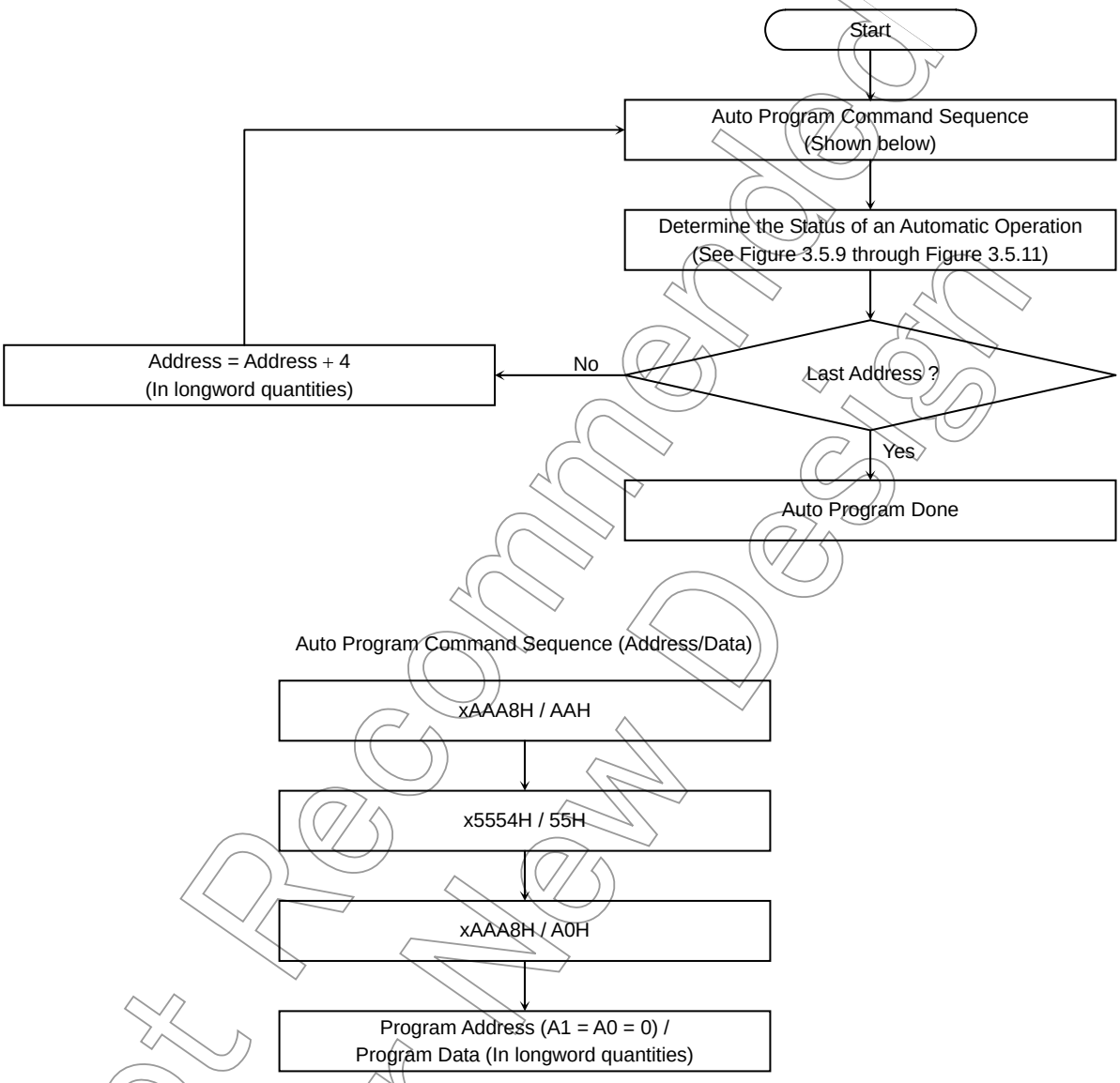


Figure 3.5.5 Auto Program Operation

Figure 3.5.6 shows flowcharts of the Auto Chip Erase and Auto Block Erase operations. Operation flows of these algorithms are the same except the command sequences. For details on the step "Determine the Status of an Automatic Operation," see Figure 3.5.9, Data Polling (DQ7) Algorithm, Figure 3.5.10, Toggle Bit (DQ6) Algorithm, and Figure 3.5.11, R/BSY Flag Algorithm.

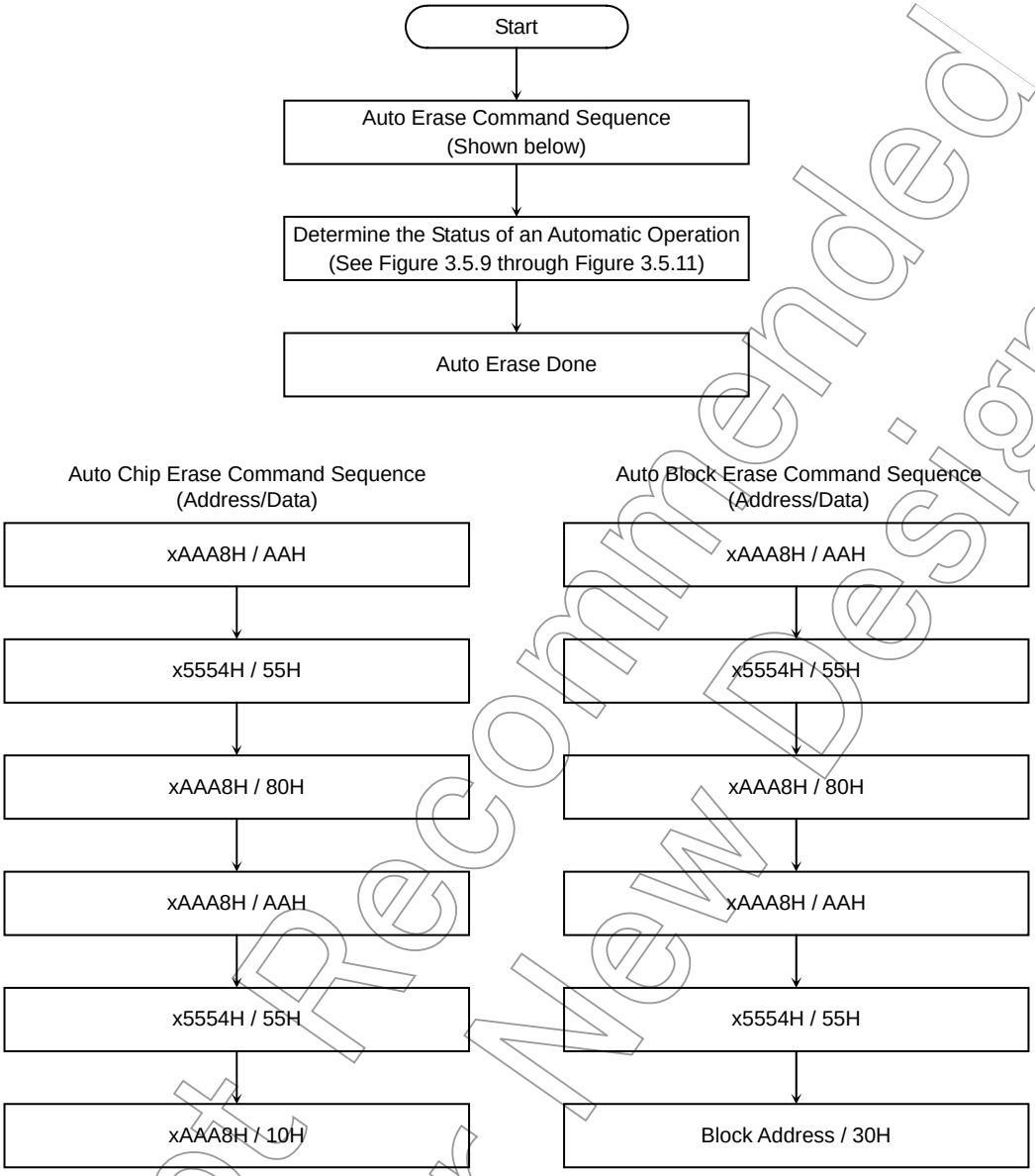


Figure 3.5.6 Auto Erase Operations

Figure 3.5.7 shows a flowchart of the Auto Block Protect operation. For details on the step "Determine the Status of an Automatic Operation," see Figure 3.5.10, Toggle Bit (DQ6) Algorithm, and Figure 3.5.11, R/BSY Flag Algorithm.

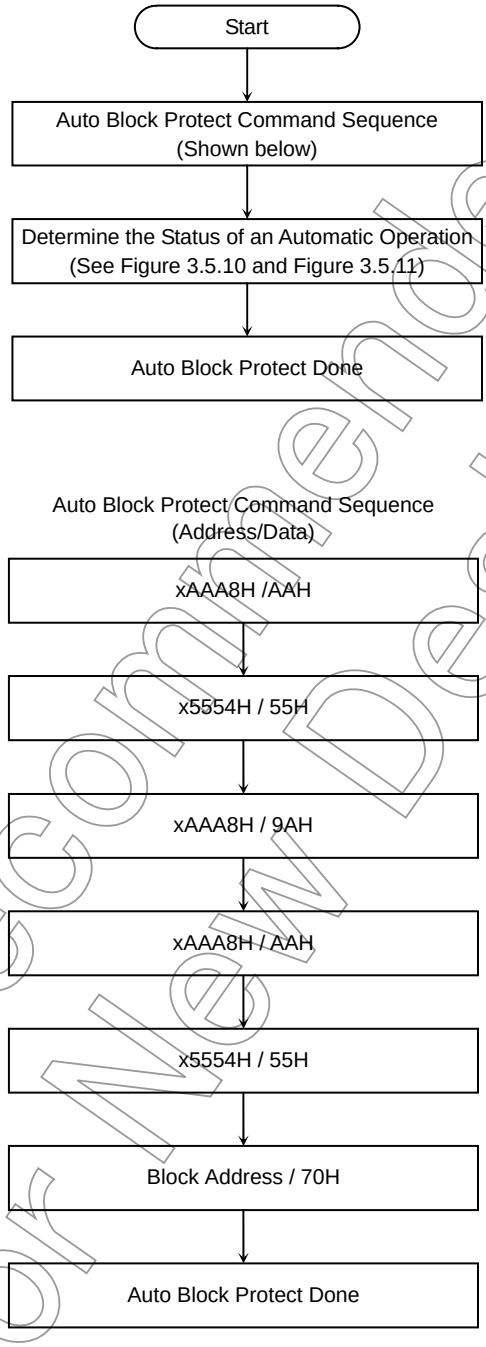


Figure 3.5.7 Auto Block Protect Operation

Figure 3.5.8 shows a flowchart of the Auto Chip Protect & Unprotect operation. For details on the step "Determine the Status of an Automatic Operation," see Figure 3.5.9, Data Polling (DQ7) Algorithm, Figure 3.5.10, Toggle Bit (DQ6) Algorithm, and Figure 3.5.11, R/BSY Flag Algorithm.

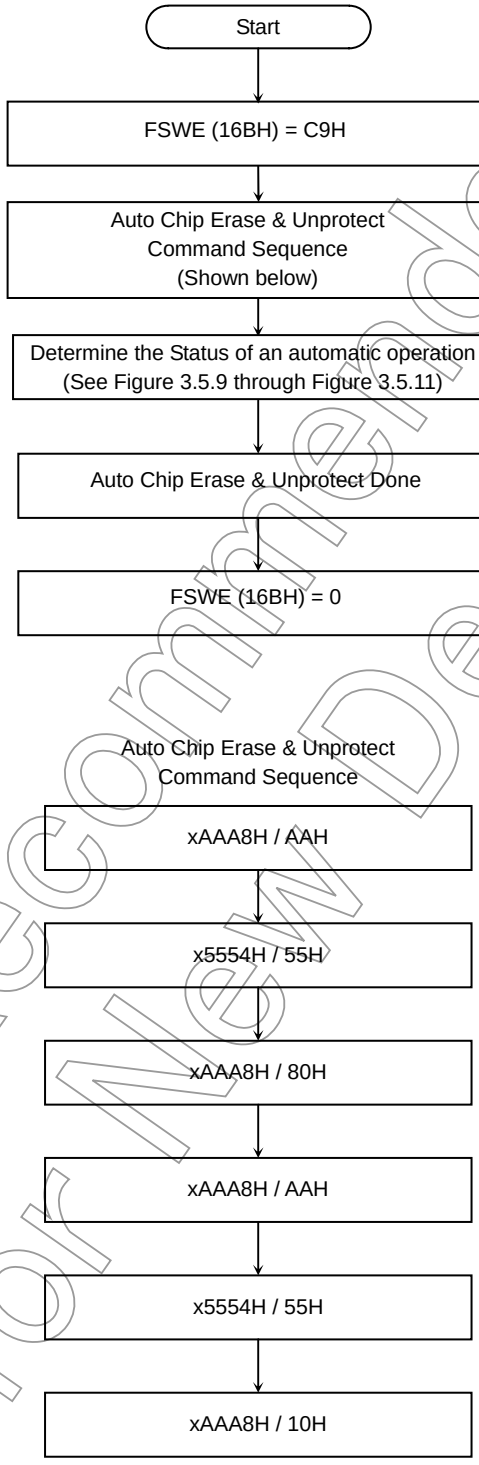
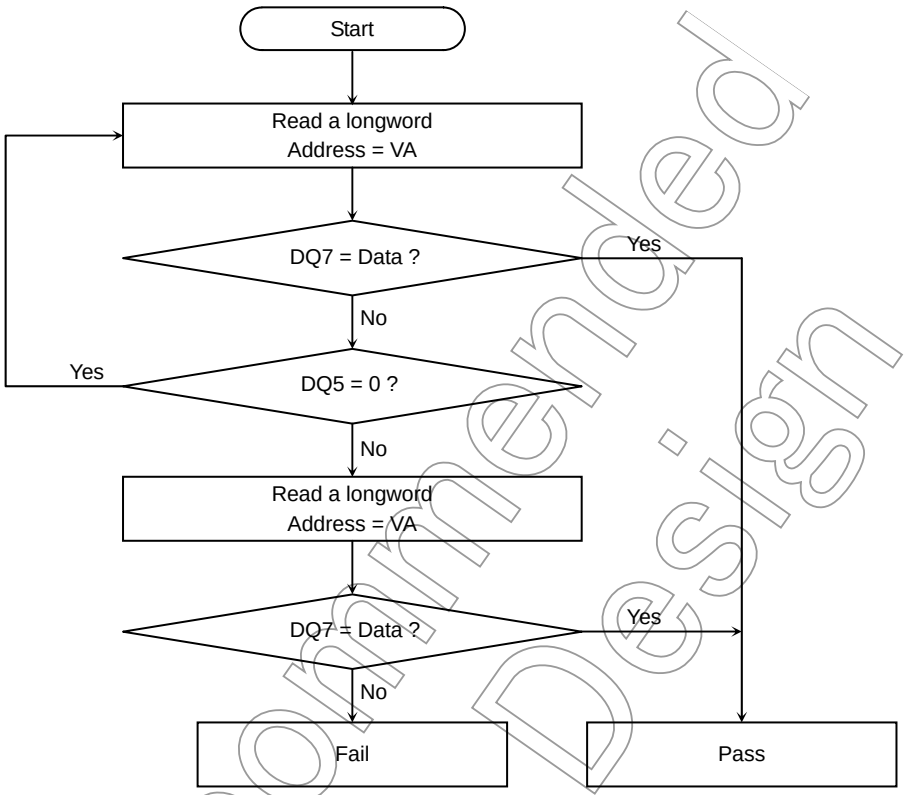


Figure 3.5.8 Auto Chip Erase & Unprotect Operation

Figure 3.5.9 shows a flowchart example of how to determine the status of an Automatic Operation by using the Data Polling (DQ7) bit.
In the Auto Block Protect operation, its operation status cannot be determined by the Data Polling (DQ7) bit. The Toggle (DQ6) and R/BSY flag bits should be checked to determine the correct status.



VA:

Addresses of the memory locations being programmed in the Auto Program operation

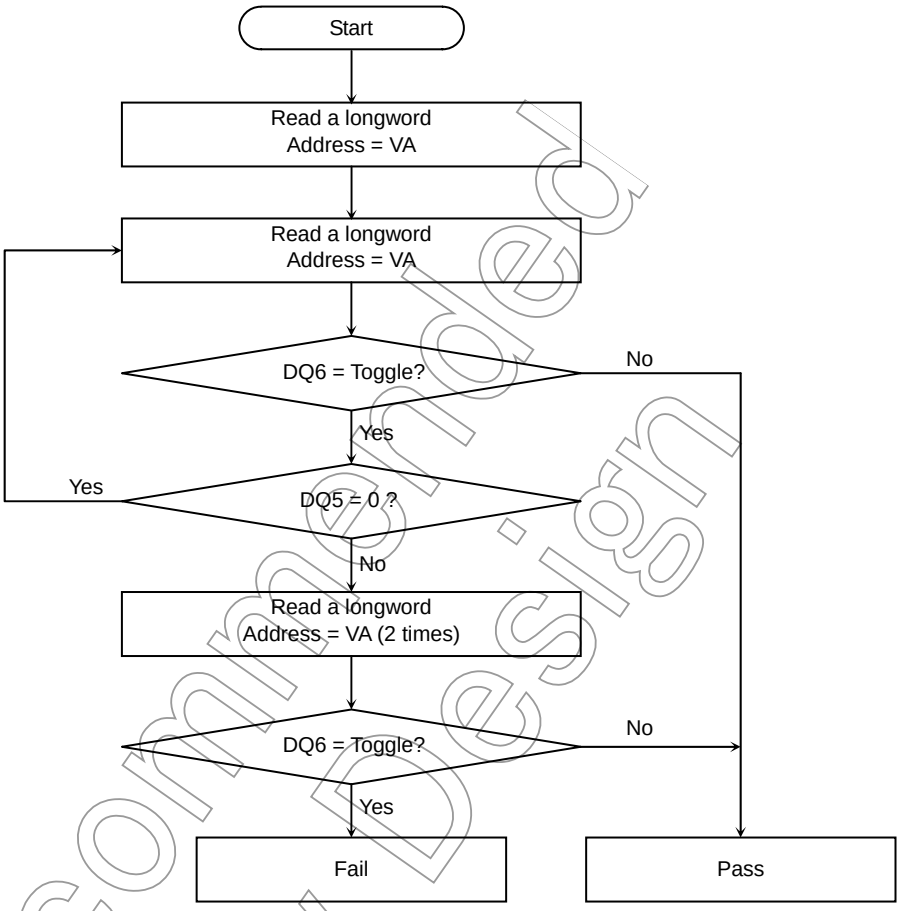
Addresses of the flash memory locations being erased in the Auto Chip Erase operation

Addresses of the memory locations in the selected block being erased in the Auto Block Erase operation

Note: DQ5 will produce a 1 if the system tries to overwrite the programmed memory locations. This indicates an incorrect usage of the flash memory, not a device failure.

Figure 3.5.9 Data Polling (DQ7) Algorithm

Figure 3.5.10 shows a flowchart example of how to determine the status of an automatic operation by using the Toggle (DQ6) bit.



VA:

Addresses of the memory locations being programmed in the Auto Program operation

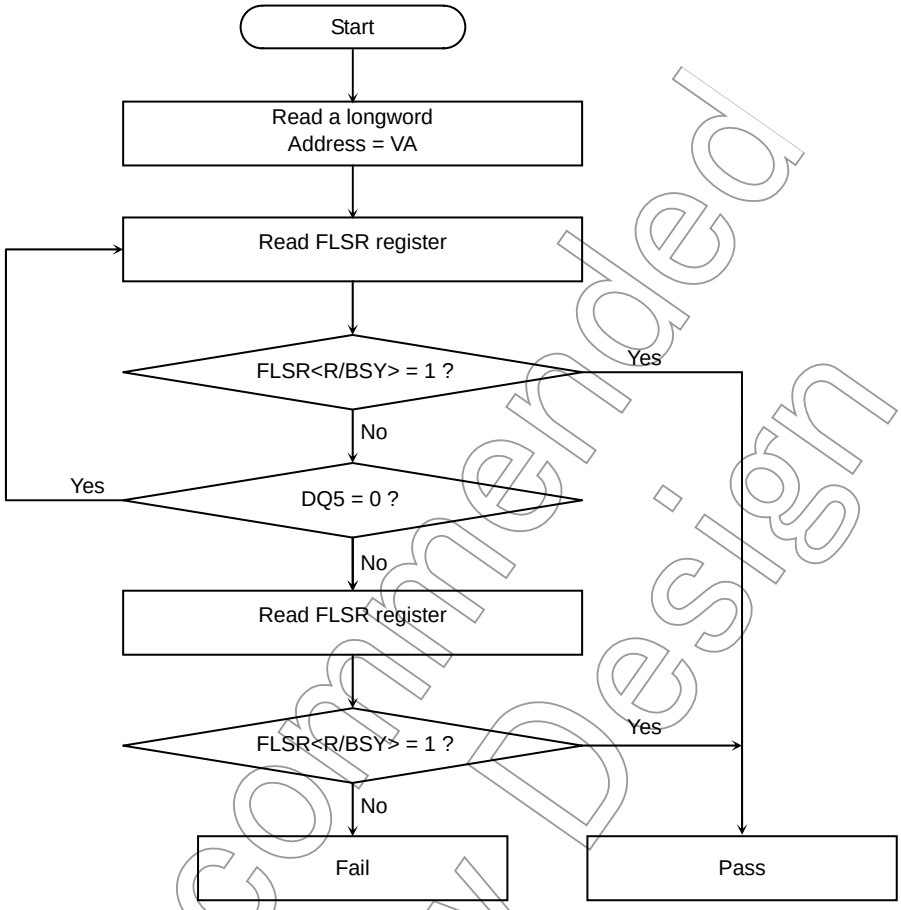
Addresses of the flash memory locations being erased in the Auto Chip Erase operation

Addresses of the memory locations in the selected block being erased in the Auto Block Erase operation

Note: DQ5 will produce a 1 if the system tries to overwrite the programmed memory locations. This indicates an incorrect usage of the flash memory, not a device failure.

Figure 3.5.10 Toggle Bit (DQ6) Algorithm

Figure 3.5.11 shows a flowchart of how to determine the status of an automatic operation by using the R/BSY flag (FLSR<R/BSY>).



VA:

Addresses of the memory locations being programmed in the Auto Program operation

Addresses of the flash memory locations being erased in the Auto Chip Erase operation

Addresses of the memory locations in the selected block being erased in the Auto Block Erase operation

Note: DQ5 will produce a 1 if the system tries to overwrite the programmed memory locations. This indicates an incorrect usage of the flash memory, not a device failure.

Figure 3.5.11 R/BSY Flag Algorithm

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

The absolute maximum ratings of a semiconductor device are a set of ratings that must not be exceeded, even for a moment. Do not exceed any of these ratings. Exceeding the rating(s) may cause the device breakdown, damage or deterioration, and may result injury by explosion or combustion. The equipment manufacturer should design so that no maximum rating value is exceeded.

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC5}	−0.5 - 6.0	V
Input voltage	V_{IN}	−0.5 - V_{CC5} + 0.5	V
Output current (total)	ΣI_{OL}	100	mA
Output current (total)	ΣI_{OH}	−100	mA
Power dissipation (Ta = 85°C)	P_D	600	mW
Soldering temperature (10 s)	T_{SOLDER}	260	°C
Storage temperature	T_{STG}	−65 - 150	°C
Operating temperature	T_{OPR}	−40 - 85	°C
Operating temperature (During flash program/erase)		0 - 70	°C
Program/erase cycles	N_{EW}	100	Cycle

Solderability of lead-free products

Test parameter	Test condition	Note
Solderability	Use of Sn-37Pb solder bath Solder bath temperature =230°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: solderability rate until forming ≥ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature =245°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	

4.2 DC Electrical Characteristics

Vcc5 = 4.5 to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	Conditions	Min	Max	Unit
Supply voltage	V_{CC5}		4.5	5.25	V
Low-level input voltage P00 - P07 (D0 to D7) PG0 - PG7 PL0 - PL3	V_{IL0}		−0.3	0.8	V
Low- level input voltage P00 - P07 (PORT) P40 - P47	V_{IL1}		−0.3	0.3∗VCC5	V
Low-level input voltage INT0 $\overline{\text{NMI}}$ $\overline{\text{RESET}}$ P70, P71, P73 - P75 PC0 - PC5 PD0 - PD7 PF0 - PF7 PM0 - PM4	V_{IL2}		−0.3	0.25∗VCC5	V
P72, PN0 - PN6	V_{IL6}		−0.3	0.3∗VCC5	V
Low-level input voltage AM0 - AM1 TEST0 - TEST1	V_{IL3}		−0.3	0.3	V
Low- level input voltage X1, XT1 (Crystal)	V_{IL4}	* Vcc3 = 3.3 V	−0.3	0.2∗VCC3	V
Low-level input voltage XT1 (CR)	V_{IL5CR}	* Vcc3 = 3.3 V	−0.3	0.2∗VCC3	V
High-level input voltage P00 - P07 (D0 to D7) PG0 - PG7 PL0 - PL3	V_{IH0}		2.2	VCC5+0.3	V
High-level input voltage P00 - P07 P40 - P47	V_{IH1}		0.7∗VCC5	VCC5+0.3	V
High-level input voltage INT0 $\overline{\text{NMI}}$ $\overline{\text{RESET}}$ P70, P71, P73 - P75 PC0 - PC5 PD0 - PD7 PF0 - PF7 PM0 - PM4	V_{IH2}		0.75∗VCC5	VCC5+0.3	V
P72, PN0 - PN6	V_{IH6}		0.7∗VCC5	VCC5+0.3	V
High-level input voltage AM0 - AM1 TEST0 - TEST1	V_{IH3}		VCC5−0.3	VCC5+0.3	V
High-level input voltage X1, XT1 (Crystal)	V_{IH4}	* Vcc3 = 3.3 V	0.8∗VCC3	VCC3+0.3	V
High-level input voltage XT1 (CR)	V_{IH5CR}	* Vcc3 = 3.3 V	0.7∗VCC3	VCC3+0.3	V

Vcc5 = 4.5 to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

Parameter	Symbol	Conditions		Min	Max	Unit
Low-level output voltage	V_{OL}	I_{OL} = 3.0 mA			0.4	V
High-level output voltage	V_{OH0}	I_{OH} = -400 μA		2.4		V
	V_{OH1}	I_{OH} = -100 μA		0.75*VCC5		
	V_{OH2}	I_{OH} = -20 μA		0.9*VCC5		
	V_{OHn}	I_{OH} = -200 μA, PF6(TX) pin only		0.82*VCC5		
Input leakage current	I_{LI}	0.0 ≦ Vin ≦ VCC5, Vin: Input voltage		0.02 (typ.)	±5	μA
Output leakage current	I_{LO}	0.2 ≦ Vin ≦ VCC5-0.2, Vin: Input voltage		0.05 (typ.)	±10	μA
Operating current (Single chip) (Note 1)	I_{CC5}	V_{CC5} = 5.25 V, X1 = 10 MHz (fc = 20 MHz)		80(typ.)	100	mA
Operating current (Standby) (Note 2)	$I_{CC5IDLE2}$	IDLE2 Mode	V_{CC5} = 5.25 V, X1 = 10 MHz (fc = 20 MHz)		90	mA
	$I_{CC5IDLE1}$	IDLE1 Mode	V_{CC5} = 5.25 V, X1 = 10 MHz (fc = 20 MHz)		30	
	$I_{CC5IDLE3}$	IDLE3 Mode	V_{CC5} = 5.25 V, Ta = -40 to 85°C V_{CC5} = 5.25 V, Ta = -10 to 55°C		220 140	μA
	$I_{CC5STOP}$	STOP Mode	V_{CC5} = 5.25 V, Ta = -40 to 85°C V_{CC5} = 5.25 V, Ta = -10 to 55°C		200 120	μA
Standby voltage (The voltage required to maintain the status of internal storage element, such as registers and RAM.)	V_{STB5}	$V_{CC3} < V_{CC5}$, $V_{IH1} < V_{CC5}$, $V_{IH2} < V_{CC5}$, $V_{IH3} < V_{CC5}$		3.0	5.25	V
Pull-up resistor	R_{RST}	$\overline{RESET}$		60	220	kΩ
	R_{CLK}	CLK				
	R_{REGEN}	REGEN				
Schmitt trigger hysteresis	V_{TH}	INT0, $\overline{NMI}$, $\overline{RESET}$, P70 - P75, PC0 - PC5, PD0 - PD7, PF0 - PF7, PM0 - PM4, PN0 - PN6		0.4	1.0 (typ.)	V

Note 1: Value when the external bus is not operating

Note 2: The $I_{CC5IDLE3}$ and $I_{CC5STOP}$ values are those measured when the supply voltage sensing circuitry of the RAM controller is not operating.

Single Boot Mode

Vcc5 = 4.5 to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C
(Ta = 0 to 70°C for program/erase operations)

Parameter	Symbol	Conditions	Min	Max	Unit
Operating current (Read)	I_{DDO1}	fc = 20 MHz	80	100	mA
Operating current (Program)	I_{DDO2}		—	100	mA
Operating current (Erase)	I_{DDO3}		—	110	mA
Standby current	I_{DDS}	V_{CC5} = 5.25 V, Ta = -40 to 85°C V_{CC5} = 5.25 V, Ta = -10 to 55°C	—	200 120	μA

4.3 AC Electrical Characteristics

Read Cycle

Vcc5 = 4.5 to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

No.	Parameter	Symbol	Min	Max	20MHz	16MHz	Unit
1	Oscillator frequency (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System clock cycle period (= T)	t_{CYC}	50	62.5	50	62.5	ns
3	CLK pulse width low	t_{CL}	0.5 × T – 15		10	16	ns
4	CLK pulse width high	t_{CH}	0.5 × T – 15		10	16	ns
5-1	A0-A23 transition to D0-D7 data in @ 0 wait state	t_{AD}		2.0 × T – 50	50	75	ns
5-2	A0-A23 transition to D0-D7 data in @ 1 wait state	t_{AD3}		3.0 × T – 50	100	138	ns
6-1	$\overline{RD}$ asserted to D0-D7 data in @ 0 wait state	t_{RD}		1.5 × T – 45	30	49	ns
6-2	$\overline{RD}$ asserted to D0-D7 data in @ 1 wait state	t_{RD3}		2.5 × T – 45	80	111	ns
7-1	$\overline{RD}$ pulse width low @ 0 wait state	t_{RR}	1.5 × T – 20		55	74	ns
7-2	$\overline{RD}$ pulse width low @ 1 wait state	t_{RR3}	2.5 × T – 20		105	136	ns
8	A0-A23 valid to $\overline{RD}$ asserted	t_{AR}	0.5 × T – 20		5	11	ns
9	$\overline{RD}$ asserted to CLK low	t_{RK}	0.5 × T – 20		5	11	ns
10	A0-A23 transition to D0-D7 hold	t_{HA}	0		0	0	ns
11	$\overline{RD}$ negated to D0-D7 hold	t_{HR}	0		0	0	ns
12	$\overline{WAIT}$ setup time	t_{TK}	15		15	15	ns
13	$\overline{WAIT}$ hold time	t_{KT}	5		5	5	ns

Write Cycle

Vcc5 = 4.5 to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

No.	Parameter	Symbol	Min	Max	20MHz	16MHz	Unit
1	Oscillator frequency (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System clock cycle period	t_{CYC}	50	62.5	50	62.5	ns
3	CLK pulse width low	t_{CL}	0.5 × T – 15		10	16	ns
4	CLK pulse width high	t_{CH}	0.5 × T – 15		10	16	ns
5-1	D0-D7 valid to $\overline{WR}$ negated @ 0 wait state	t_{DW}	1.25 × T – 35		28	43	ns
5-2	D0-D7 valid to $\overline{WR}$ negated @ 1 wait state	t_{DW3}	2.25 × T – 35		78	106	ns
6-1	$\overline{WR}$ pulse width low @ 0 wait state	t_{WW}	1.25 × T – 30		33	48	ns
6-2	$\overline{WR}$ pulse width low @ 1 wait state	t_{WW3}	2.25 × T – 30		83	111	ns
7	A0-A23 transition to $\overline{WR}$ asserted	t_{AW}	0.5 × T – 20		5	11	ns
8	$\overline{WR}$ asserted to CLK low	t_{WK}	0.5 × T – 20		5	11	ns
9	$\overline{WR}$ negated to A0-A23 hold	t_{WA}	0.25 × T – 5		8	11	ns
10	$\overline{WR}$ negated to D0-D7 hold	t_{WD}	0.25 × T – 5		8	11	ns
11	$\overline{WAIT}$ setup time	t_{TK}	15		15	15	ns
12	$\overline{WAIT}$ hold time	t_{KT}	5		5	5	ns
13	$\overline{RD}$ negated to D0-D7 out	t_{RDO}	1.25 × T – 35		20	26	ns

AC test conditions:

Output conditions of the D0 to D7, A0 to A7, A8 to A15, A16 to A23, $\overline{RD}$ and $\overline{WR}$ pins:

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

Output conditions of pins other than the above-mentioned ones:

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

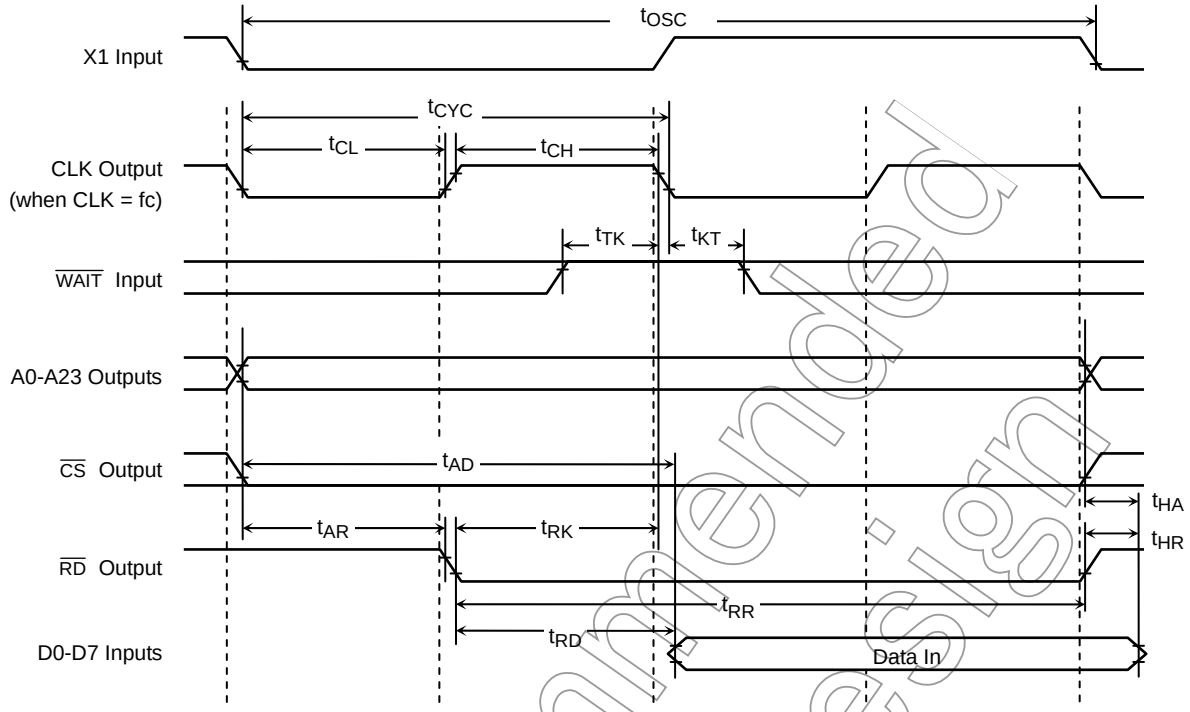
Input conditions of the P00 to P07 (D0 – D7) pins:

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

Input conditions of pins other than the above-mentioned ones:

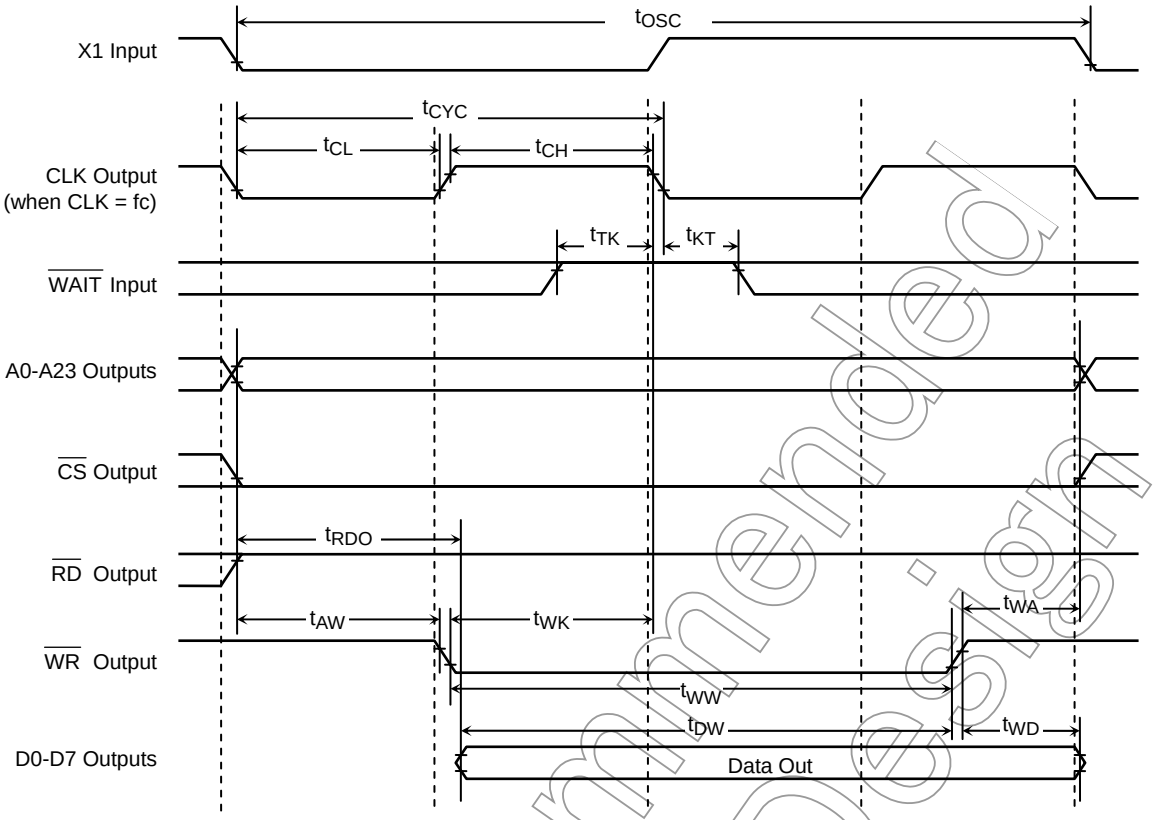
High = 0.8 × V_{CC5}, Low = 0.2 × V_{CC5}, CL = 50 pF

(1) Read Cycle Timing (0 Wait State)



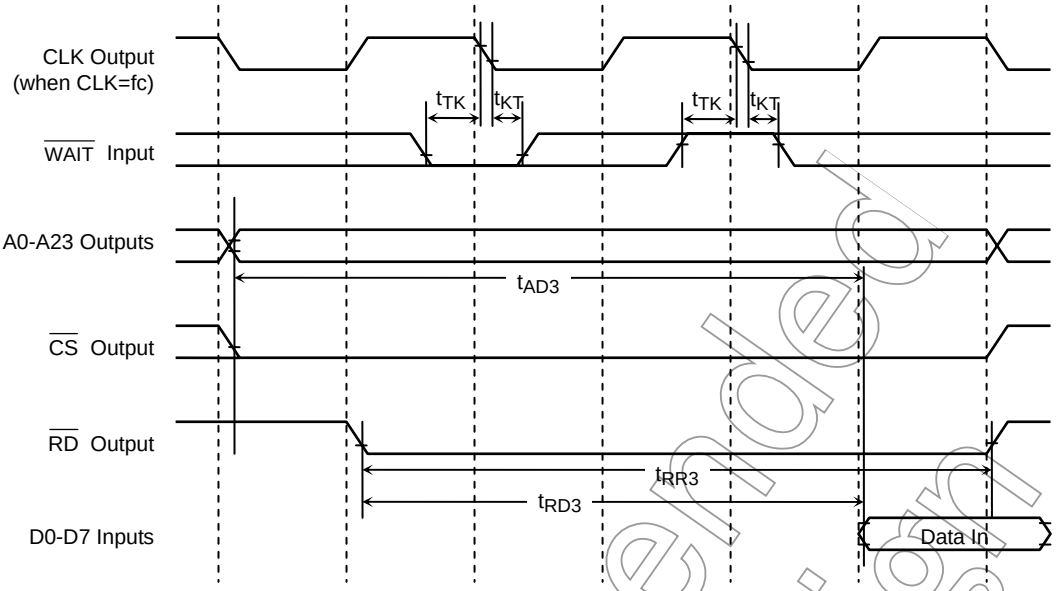
Note: The signals other than the X1 signal are derived from the X1 signal. Thus, certain timing delays occur in the generation of these signals. Since these delay times vary depending on each sample device, the phase differences between the X1 signal and the other signals cannot be specified. The phase relationship shown in the above timing diagram is only an example.

(2) Write Cycle Timing (0 Wait State)

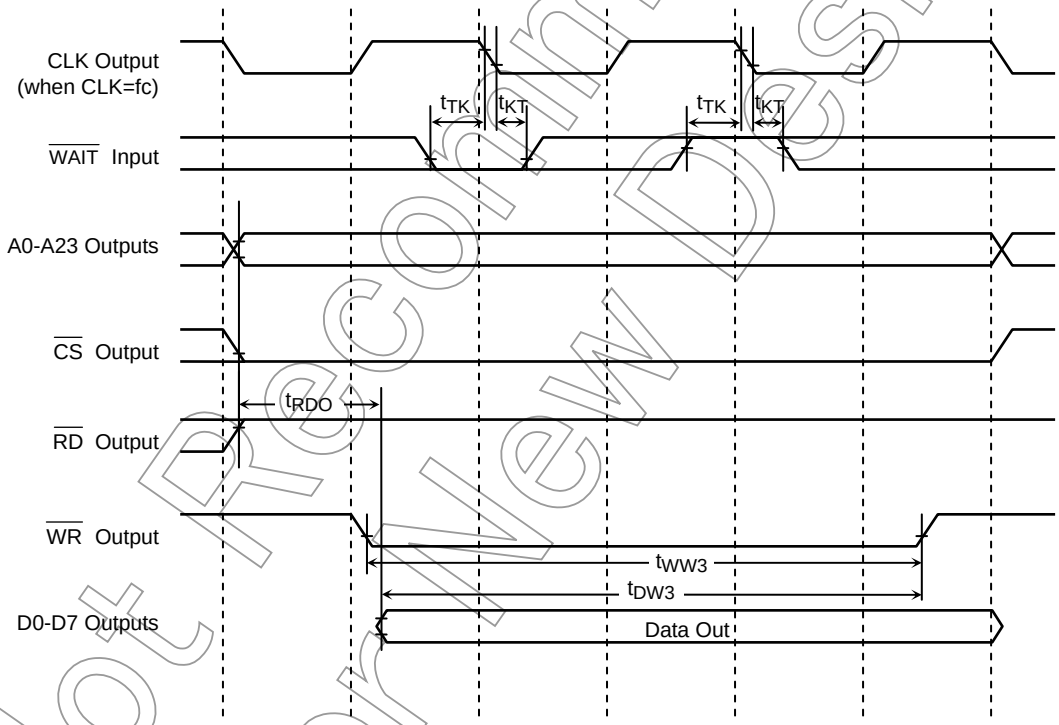


Note: The signals other than the X1 signal are derived from the X1 signal. Thus, certain timing delays occur in the generation of these signals. Since these delay times vary depending on each sample device, the phase differences between the X1 signal and the other signals cannot be specified. The phase relationship shown in the above timing diagram is only an example.

(3) Read Cycle Timing (1 Wait State)



(4) Write Cycle Timing (1 Wait State)



4.4 AD Converter Characteristics

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	Min	Typ.	Max	Unit
Analog reference voltage (+)	V_{REFH}	V_{CC5} − 0.2	V_{CC5}	V_{CC5}	V
Analog reference voltage (−)	V_{REFL}	GND	GND	GND	
Supply voltage for AD converter	AV_{CC}	V_{CC5} − 0.2	V_{CC5}	V_{CC5}	
Ground for AD converter	AV_{SS}	GND	GND	GND	
Analog input voltage	AV_{IN}	V_{REFL}		V_{REFH}	
Supply current for analog reference voltage <VREFON> = 1	I_{REF}		0.8	1.2	mA
Supply current for analog reference voltage <VREFON> = 0			0.02	5	μA
Total error (excluding quantization error)	E_T			±3.0	LSB

Note: A least significant bit (LSB) is a unit of voltage equal to the smallest resolution of the AD converter.
± 3 LSB = 3 × (V_{REFH}−V_{REFL})/1024 ≈ ±15mV (V_{REFH} = 5.0V, V_{REFL} = 0.0V)

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

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	Equation		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock cycle period	t_{VCK}	8T + 100		500		600		ns
Clock pulse width low	t_{VCKL}	4T + 40		240		290		ns
Clock pulse width high	t_{VCKH}	4T + 40		240		290		ns

4.6 Serial Channel Timing

(1) SCLK Input Mode (I/O interface mode)

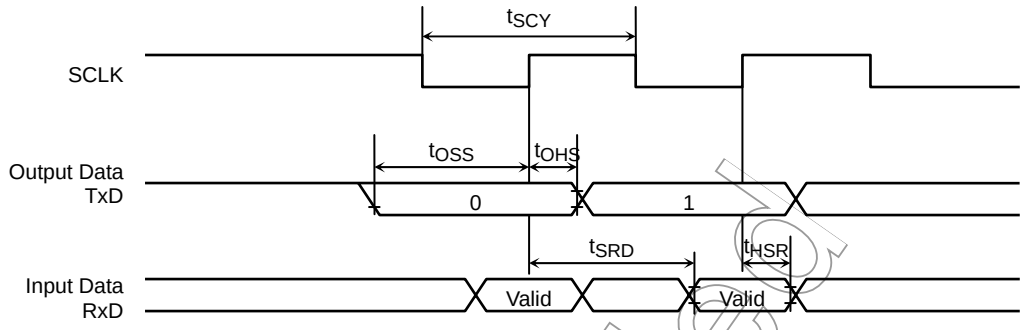
Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	Equation		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle period	t_{SCY}	16T		0.8		1.0		μs
Output data to SCLK high	t_{OSS}	t_{SCY}/2 − 4T −110		90		140		ns
SCLK high to output data hold	t_{OHS}	t_{SCY}/2 + 2T		500		625		
SCLK high to input data hold	t_{HSR}	3T+10		160		197.5		
SCLK high to valid data in	t_{SRD}		t_{SCY}		800		1000	

(2) SCLK Output Mode (I/O interface mode)

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	Equation		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle period (programmable)	t_{SCY}	16T	8192T	0.8	409.6	1.0	512	μs
Output data to SCLK high	t_{OSS}	t_{SCY}/2 − 40		360		460		ns
SCLK high to output data hold	t_{OHS}	t_{SCY}/2 − 40		360		460		
SCLK high to input data hold	t_{HSR}	0		0		0		
SCLK high to valid data in	t_{SRD}		t_{SCY} − T −180		570		757.5	



(3) SCLK Input Mode (UART mode)

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

Parameter	Symbol	Equation		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle period	T_{SCY}	4T + 20		220		270		ns
SCLK pulse width low	T_{SCYL}	2T + 5		105		130		
SCLK pulse width high	T_{SCYH}	2T + 5		105		130		

4.7 Interrupt Operation

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

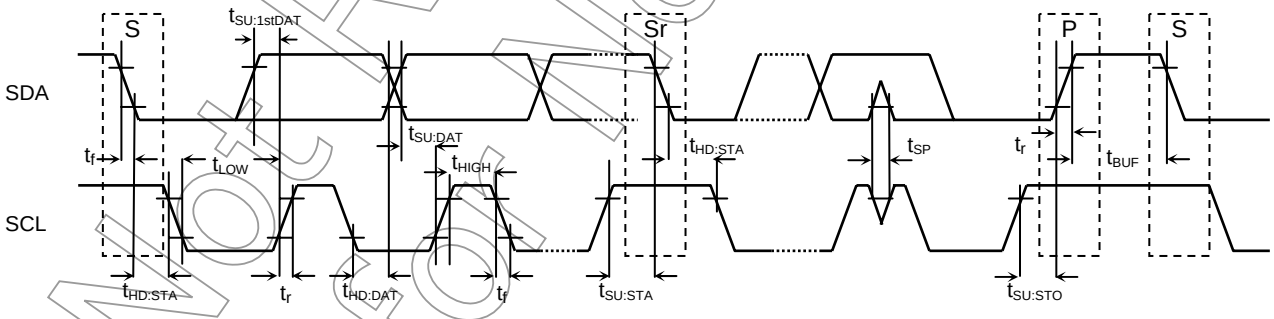
Parameter	Symbol	Equation		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 pulse width low	t_{INTAL}	4T		200		250		ns
$\overline{NMI}$, INT0 pulse width high	t_{INTAH}	4T		200		250		
INT1-INT7 pulse width low	t_{INTBL}	8T + 100		500		600		
INT1-INT7 pulse width high	t_{INTBH}	8T + 100		500		600		

4.8 Serial Bus Interface

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	fc = 20 MHz						Unit
		400 KHz <SCK3:0>=1000		100 KHz <SCK3:0>=1111		(Note2) <SCK3:0>=0011 ~ 0110		
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	f_{SCL}	0	400	0	100	0	fc/(2^n+8)	KHz
Hold time for repeated START condition (After this period, the first clock pulse is generated.)	$t_{HD:STA}$	650		4500		2^{n-1}/fc		ns
Low period of the SCL clock	t_{LOW}	1300		4700		2^{n-1}/fc		
High period of the SCL clock	t_{HIGH}	600		4000		(2^{n-1}+8)/fc		
Setup time for repeated START condition	$t_{SU:STA}$	Software		Software		Software		
Data hold time	$t_{HD:DAT}$	0	900	0	3450	0	6/fc	
Data setup time	$t_{SU:DAT}$	100		250		(2^{n-1}−6)/fc		
Data setup time (After transfer 1st data bit)	$t_{SU:1stDAT}$	100		250		(2^{n-1}−12)/fc		
Rise time of both SDA and SCL signals	t_r		300 (Reception)		1000 (Reception)			
Fall time of both SDA and SCL signals	t_f		300		300			
Setup time for a STOP condition	$t_{SU:STO}$	950		4200		(2^{n-1}+12)/fc		
Bus free time between a STOP and START condition	t_{BUF}	Software		Software		Software		
Capacitive load for each bus line	C_b		400		400		400	pF
Noise margin at the Low level for each connected device (including hysteresis)	V_{nL}	0.2∗VCC5		0.2∗VCC5		0.2∗VCC5		V
Noise margin at the High level for each connected device (including hysteresis)	V_{nH}	0.2∗VCC5		0.2∗VCC5		0.2∗VCC5		V
Pulse width of spikes suppressed by the input filter	t_{SP}	0	50	n/a	n/a	n/a	n/a	ns

Note 1: The above values are referred to V_{IHmin} and V_{ILmax}.
Note 2: Value for setting <SCK 3:0> = 0011 - 0110 (n = 8 - 11) includes tf and tr.
Note 3: n/a = not applicable



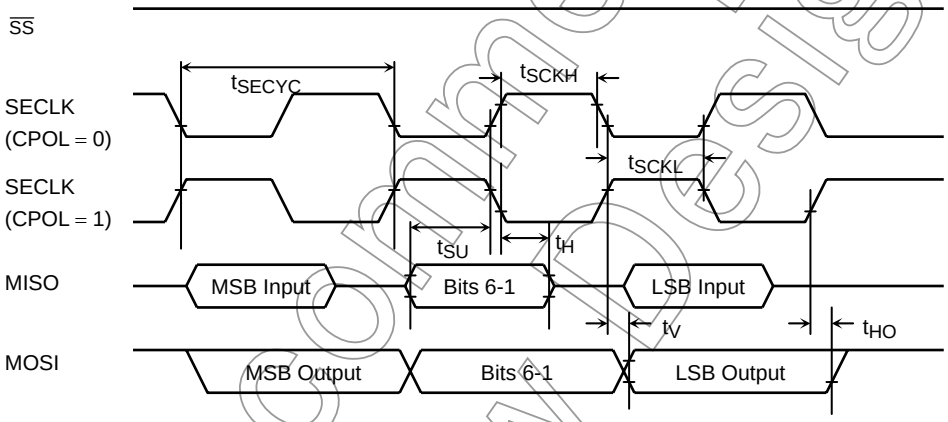
S: START P: STOP Sr: RESTART

4.9 Serial Expansion Interface

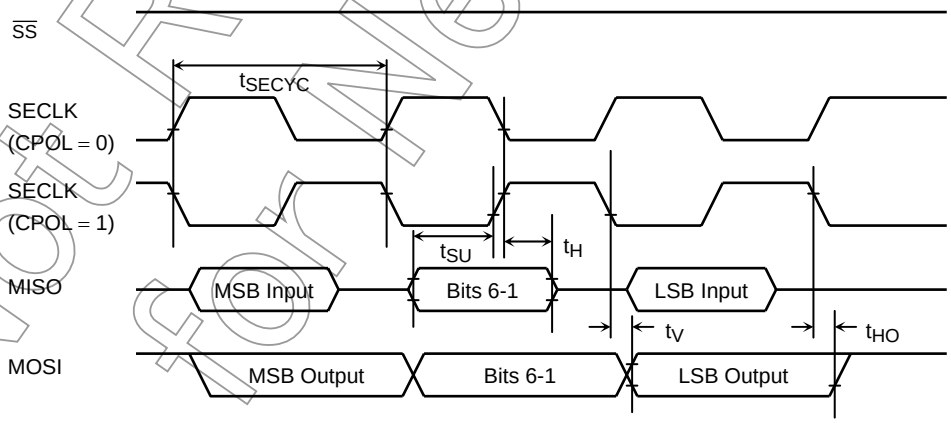
Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	Equation		20 MHz		Unit
		Min	Max	Min	Max	
SECLK cycle period	t_{SECYC}	5T	40T	250	2000	ns
$\overline{SS}$ asserted to SECLK	t_{LEAD}	4T		200		ns
SECLK to $\overline{SS}$ negated	t_{LAG}	4T		200		ns
SECLK pulse width high	t_{SCKH}	t_{SECLK}/2-9		116		ns
SECLK pulse width low	t_{SCKL}	t_{SECLK}/2-9		116		ns
Input data setup	t_{SU}	t_{SECLK}/4-10		52		ns
Input data hold	t_H	t_{SECLK}/4		62		ns
Output data valid	t_V		t_{SECLK}/4		62	ns
Output data hold	t_{HO}	0		0		ns

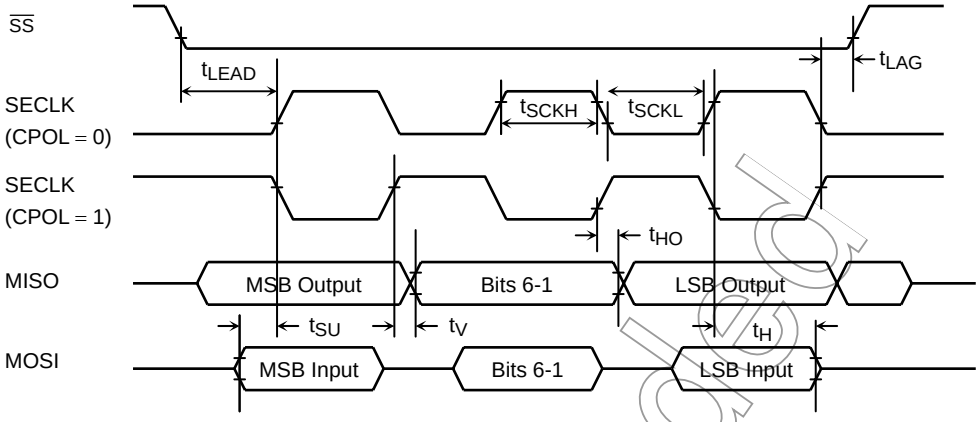
a) SEI Master (CPHA = 0)



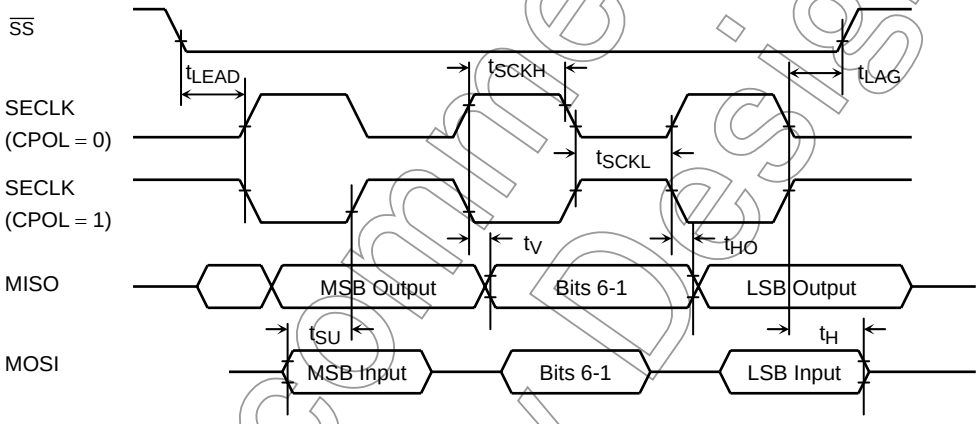
b) SEI Master (CPHA = 1)



c) SEI slave (CPHA = 0)



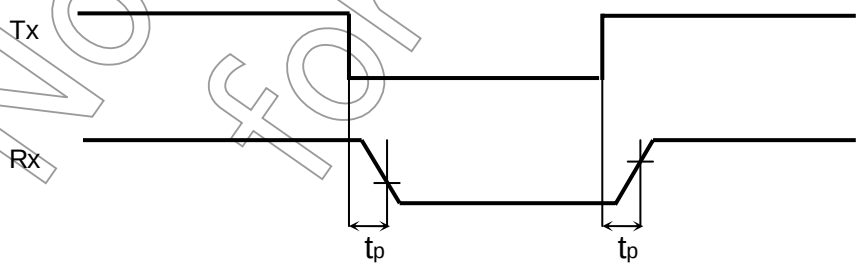
d) SEI slave (CPHA = 1)



4.10 CAN Controller

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

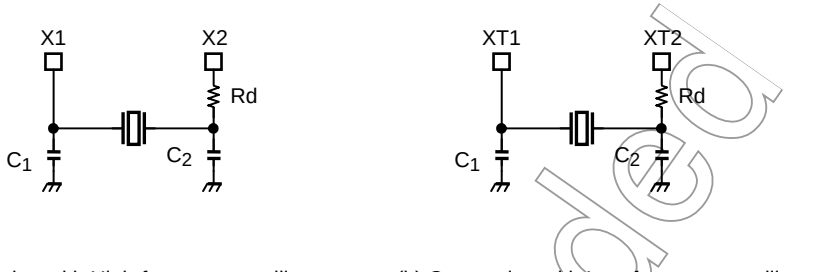
Symbol	Parameter	Equation		20 MHz		unit
		Min	Max	Min	Max	
tcclk	CAN clock period	2T		100		ns
tp	Tx low/high to Rx low/high		2tcclk-20		180	ns



4.11 Recommended Oscillator Circuit

Recommended oscillator circuits for the TMP92FD54AI are shown below:

(1) Oscillator circuit examples



(a) Connection with High-frequency oscillator (b) Connection with Low-frequency oscillator

Figure 4.11.1 Oscillation Circuits

Note: The load capacitance on the oscillator connection pins is the sum of C1 and C2 in the oscillator circuit (or incorporated in a resonator) and stray board capacitance. Since the total load capacitance varies with the board layout, the resonator might fail to work properly. To prevent this problem, the board traces near the oscillator circuit should be as short as possible. It is recommended to evaluate the oscillator using the actual application board.

(2) Recommended ceramic resonator circuit

The TMP92FD54AI high-frequency oscillator circuit has been evaluated by Murata Manufacturing Co., Ltd. For details, please contact your Murata representative.

Figure 4.11.1 shows the recommended circuit constants for the ceramic resonator manufactured by Murata.

Figure 4.11.1 Recommended ceramic resonator for the TMP92FD54AI (manufactured by Murata)

Oscillation Frequency [MHz]	Resonator Part Number		Recommended Constant				TMP92FD54AI Operating Conditions	
			C1 [pF] (Note 1)	C2 [pF] (Note 1)	Rf [Ω]	Rd [Ω]	Supply Voltage [V]	Temperature [°C]
8.0	SMD	CSTCE8M00G15C()–R0	(33)	(33)	Open	330	4.5 to 5.5	–40 to 85
10.0	SMD	CSTCE10M0G15C()–R0	(33)	(33)	Open	330		

Note 1: Enclosed in parentheses are the built-in load capacitor values.

Note 2: Part numbers and specifications of resonators manufactured by Murata are subject to change without notice. For details, please visit Murata's website at http://www.murata.co.jp.

4.12 Voltage Regulator

Voltage Regulator

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

Parameter		Symbol	Conditions	Min	Typ.	Max	Unit
Input voltage		Vcc5	Including ripple voltage (Vp-p).	4.5	5.0	5.25	V
	Peak-to-Peak voltage (ripple voltage)	Vp-p	Ripple frequency ≤ 100 Hz (sine wave)	–	0	0.75	V
			Ripple frequency > 100 Hz (sine wave)	–	0	0.3	V
			All ripple frequencies (except sine wave)	–	0	0.2	V
Output voltage		REGOUT	4.5 ≤ Vin ≤ 5.25, ILoad = 100 mA (Vin = Vcc5) Ta = –40 to 85°C	3.0	3.3	3.6	V
Output current		Iro	Vin – REGOUT = 1.0 V Ta = –40 to 85°C	0	–	150	mA
Quiescent current		Iq	ILoad ≤ 10 μA, Ta = –40 to 85°C	–	–	80	μA
		Iq1	10 μA < ILoad < 100 mA, Ta = 25°C	–	–	800	μA
		Iop	ILoad = 150 mA, Ta = –40 to 85°C	–	–	10	mA
Standby current		Is	REGEN = 0 (regulator only)	–	0.1	0.2	μA

0.5 [Ω] ≤ ESR ≤ 5.0 [Ω]

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Stabilizing capacitor	Cs	Cb = 10 μF, ESR = 4.7 Ω	0.1	–	10	μF
Bypass capacitor	Cb	Cs = 10 μF, ESR = 4.7 Ω (Cs ≥ Cb)	0.1	–	10	μF
Input capacitor	Cin (Note)	Cs = 10 μF, ESR = 4.7 Ω	4.7	–	22	μF
Stabilizing resistor	ESR	Cs = 10 μF, Cb = 0.1 μF	0.5	–	5	Ω

Note: Tantalum capacitors are recommended.

0.5 [Ω] ≤ ESR ≤ 50 [Ω]

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Stabilizing capacitor	Cs	Cb = 0.6 μF, ESR = 47 Ω	0.1	–	10	μF
Bypass capacitor	Cb	Cs = 10 μF, ESR = 47 Ω (Cs ≥ Cb)	0.6	–	10	μF
Input capacitor	Cin (Note)	Cs = 10 μF, ESR = 47 Ω	4.7	–	22	μF
Stabilizing resistor	ESR	Cs = 10 μF, Cb = 0.6 μF	0.5	–	50	Ω

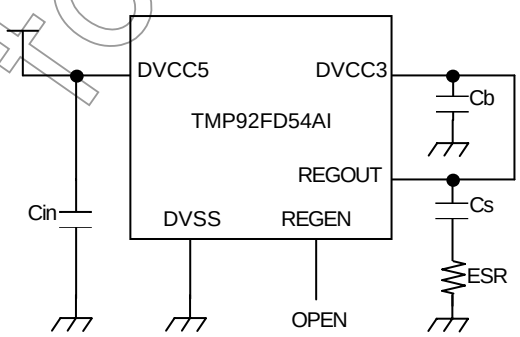
Note: Tantalum capacitors are recommended.

0.5 [Ω] ≤ ESR ≤ 100 [Ω]

Vcc5 = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = –40 to 85°C

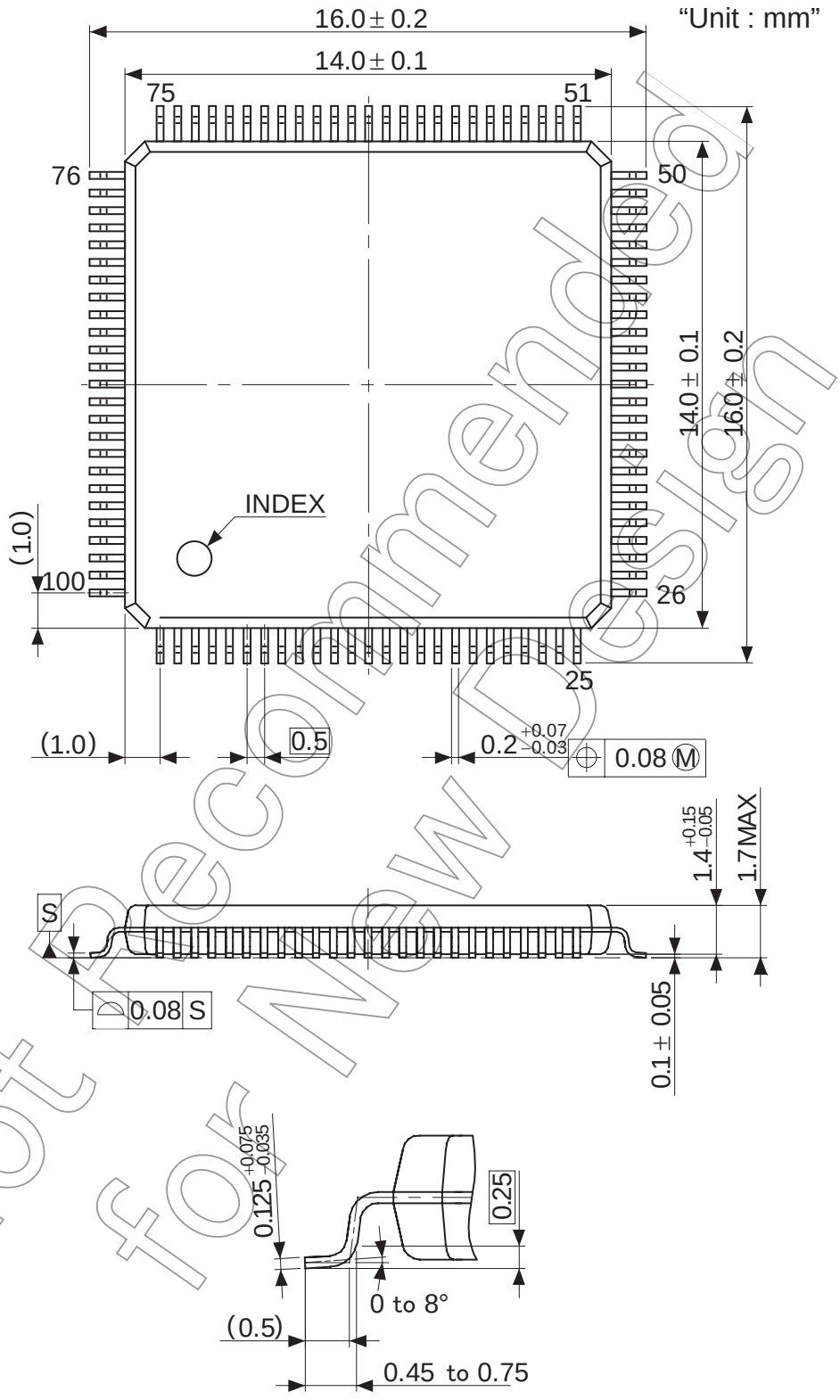
Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Stabilizing capacitor	Cs	Cb = 1.0 μF, ESR = 100 Ω	0.1	–	10	μF
Bypass capacitor	Cb	Cs = 10 μF, ESR = 100 Ω (Cs ≥ Cb)	1.0	–	10	μF
Input capacitor	Cin (Note)	Cs = 10 μF, ESR = 100 Ω	4.7	–	22	μF
Stabilizing resistor	ESR	Cs = 10 μF, Cb = 1.0 μF	0.5	–	100	Ω

Note: Tantalum capacitors are recommended.



5. Package Dimensions

Package Dimensions：LQFP100-P-1414-0.50H



Note1: The drawings shown may not accurately represent the actual shape or dimensions.

RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively “TOSHIBA”), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively “Product”) without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA’s written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact (“Unintended Use”). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.