# Toshiba Semiconductor and Storage - TMP91FW27UG Datasheet

**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

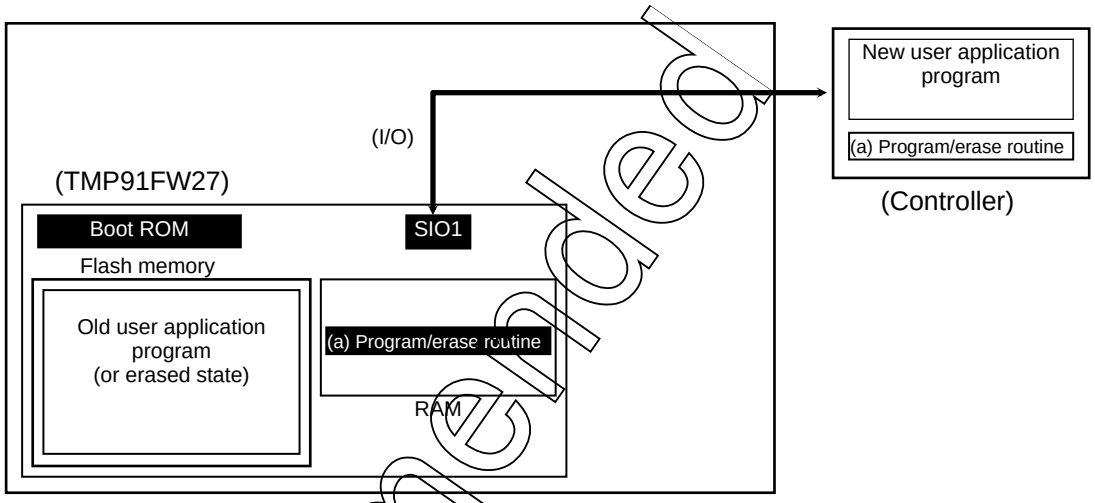
| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 900/L1 |
| Core Size | 16-Bit |
| Speed | 27MHz |
| Connectivity | EBI/EMI, I²C, IrDA, UART/USART |
| Peripherals | DMA, WDT |
| Number of I/O | 53 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 12K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.2V ~ 3.6V |
| Data Converters | A/D 4x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP |
| Supplier Device Package | 64-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmp91fw27ug |

Table 2.2.2 Pin Names and Functions (2/2)

| Pin Names | Number of Pins | I/O | Functions |
|---|---|---|---|
| P73<br>TA4IN | 1 | I/O<br>Input | Port 73: I/O port<br>8-bit timer 4 Input: Input pin of 8-bit timer TMRA4 |
| P74<br>TA5OUT | 1 | I/O<br>Output | Port 74: I/O port<br>8-bit timer 5 output: Output pin of 8-bit timer TMRA4 or TMRA5 |
| P80<br>TB0IN0<br>INT5 | 1 | I/O<br>Input<br>Input | Port 80: I/O port<br>16-bit timer 0 Input 0: Input of count/capture trigger in 16-bit timer TMRB0<br>Interrupt request pin 5: Interrupt request pin with selectable rising/falling edge |
| P81<br>TB0IN1<br>INT6 | 1 | I/O<br>Input<br>Input | Port 81: I/O port<br>16-bit timer 0 Input 1: Input of count/capture trigger in 16-bit timer TMRB0<br>Interrupt request pin 6: Interrupt request pin of rising edge |
| P82<br>TB0OUT0 | 1 | I/O<br>Output | Port 82: I/O port<br>16-bit timer 0 output 0: Outpit pin of 16-bit timer TMRB0 |
| P83<br>TB0OUT1 | 1 | I/O<br>Output | Port 83: I/O port<br>16-bit timer 0 output 1: Output pin of 16-bit timer TMRB0 |
| P90<br>TXD0 | 1 | I/O<br>Output | Port 90: I/O port<br>Serial 0 send data: Open-drain output pin by programmable |
| P91<br>RXD0 | 1 | I/O<br>Input | Port 91: I/O port<br>Serial 0 receive data |
| P92<br>SCLK0<br>$\overline{\text{CTS0}}$ | 1 | I/O<br>I/O<br>Input | Port 92: I/O port<br>Serial 0 clock I/O<br>Serial 0 data send enable (Clear to send) |
| P93<br>TXD1 | 1 | I/O<br>Output | Port 93: I/O port<br>Serial 1 send data: Open-drain output pin by programmable |
| P94<br>RXD1 | 1 | I/O<br>Input | Port 94: I/O port<br>Serial 1 receive data |
| P95<br>SCLK1<br>$\overline{\text{CTS1}}$ | 1 | I/O<br>I/O<br>Input | Port 95: I/O port<br>Serial 1 clock I/O<br>Serial 1 data send enable (Clear to send) |
| P96<br>XT1 | 1 | I/O<br>Input | Port 96: I/O port. Open-drain output pin.<br>Low frequency oscillator connection pin |
| P97<br>XT2 | 1 | I/O<br>Output | Port 97: I/O port. Open-drain output pin.<br>Low frequency oscillator connection pin |
| ALE | 1 | Output | Address latch enable (It can be set as prohibition of an output for noize reduction.) |
| $\overline{\text{NMI}}$ | 1 | Input | Non-Maskable interrupt request pin: Interrupt request pin with programmable falling edge level or with both edge levels programmable (Schmitt input). |
| AM0 and AM1 | 2 | Input | Operation mode:<br>Fixed to AM1 = “1” and AM0 = “1”. |
| $\overline{\text{RESET}}$ | 1 | Input | Reset: Initialize LSI. (Schmitt input, with pull-up resistor) |
| AVCC | 1 | | Pin used to both power supply pin for AD converter and standard power supply for AD converter (H). |
| AVSS | 1 | | |

*(Step-3) Copying the program/erase routine to the RAM*

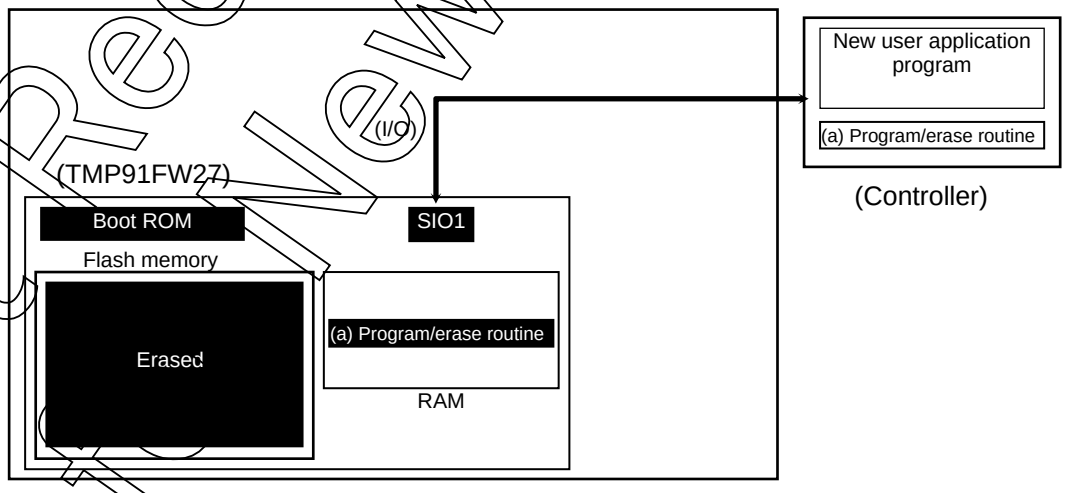
After password verification is completed, the boot ROM copies the program/erase routine (a) from the controller to the RAM using serial communications. The program/erase routine must be stored within the RAM address range of 001000H to 003DFFH.




*(Step-4) Executing the program/erase routine in the RAM*

Control jumps to the program/erase routine (a) in the RAM. If necessary, the old user application program is erased (sector erase or chip erase).

Note: The boot ROM is provided with an erase command, which enables the entire chip to be erased from the controller without using the program/erase routine. If it is necessary to erase data on a sector basis, incorporate the necessary code in the program/erase routine.




Not Recommended for New Design

#### 3.2.4.6 Data Transfer Formats

Table 3.2.7 to Table 3.2.13 show the operation command data and the data transfer format for each operation mode.

Table 3.2.7 Operation Command Data

| Operation Command Data | Operation Mode |
|---|---|
| 10H | RAM Transfer |
| 20H | Flash Memory SUM |
| 30H | Product Information Read |
| 40H | Flash Memory Chip Erase |
| 60H | Flash Memory Protect Set |

Not Recommended for New Design

Table 3.2.11 Transfer Format of Single Boot Program [Product Information Read] (2/2)

| | Transfer Byte Number | Transfer Data from Controller to Device | Baud Rate | Transfer Data from Device to Controller |
|---|---|---|---|---|
| Boot ROM | 61st byte to 64th byte | | | Size (in half words) of flash memory sectors of the same size (4 bytes)<br>00H, 08H, 00H, 00H (from 61st byte) |
| | 65th byte | | | Number of flash memory sectors of the same size (1 byte)<br>20H |
| | 66th byte | | | CHECKSUM value for 5th to 65th bytes |
| | 67th byte | (Wait for the next operation command data) | | |

**Note 1: For the desired baud rate setting, see Table 3.2.6.**
**Note 2: After sending an error response, the device waits for operation command data (3rd byte).**

Not Recommended for New Design

#### 3.2.4.8 RAM Transfer Command (See Table 3.2.8)

1. From the controller to the device

   The data in the 1st byte is used to determine the baud rate. The 1st byte is transferred with receive operation disabled (SC1MOD0<RXE> = 0). (The baud rate is determined using an internal timer.)

   - To communicate in UART mode

     Send the value 86H from the controller to the target board using UART settings at the desired baud rate. If the serial operation mode is determined as UART, the device checks to see whether or not the desired baud rate can be set. If the device determines that the desired baud rate cannot be set, operation is terminated and no communications can be established.

2. From the device to the controller

   The data in the 2nd byte is the ACK response returned by the device for the serial operation mode setting data sent in the 1st byte. If the data in the 1st byte is found to signify UART and the desired baud rate can be set, the device returns 86H.

   - Baud rate determination

     The device determines whether or not the desired baud rate can be set. If it is found that the baud rate can be set, the boot program rewrites the BR1CR and BR1ADD values and returns 86H. If it is found that the desired baud rate cannot be set, operation is terminated and no data is returned. The controller sets a time-out time (5 seconds) after it has finished sending the 1st byte. If the controller does not receive the response (86H) normally within the time-out time, it should be considered that the device is unable to communicate. Receive operation is enabled (SC1MOD0<RXE> = 1) before 86H is written to the transmission buffer.

3. From the controller to the device

   The data in the 3rd byte is operation command data. In this case, the RAM Transfer command data (10H) is sent from the controller to the device.

4. From the device to the controller

   The data in the 4th byte is the ACK response to the operation command data in the 3rd byte. First, the device checks to see if the received data in the 3rd byte contains any error. If a receive error is found, the device returns the ACK response data for communications error (bit 3) x8H and waits for the next operation command data (3rd byte). The upper four bits of the ACK response data are undefined (They are the upper four bits of the immediately preceding operation command data).

   Next, if the data received in the 3rd byte corresponds to one of the operation commands given in Table 3.2.7, the device echoes back the received data (ACK response for normal reception). In the case of the RAM Transfer command, if read or write protection is not applied, 10H is echoed back and then execution branches to the RAM transfer processing routine. If protection is applied, the device returns the corresponding ACK response data (bit 2/1) x6H and waits for the next operation command data (3rd byte). The upper four bits of the ACK response data are undefined. (They are the upper four bits of the immediately preceding operation command data.)

   After branching to the RAM transfer processing routine, the device checks the data in the password area. For details, see 3.2.4.15 “Password”.

Not Recommended for New Design

#### 3.2.4.10 Product Information Read command (See Table 3.2.10 and Table 3.2.11)

1. The data in the 1st and 2nd bytes is the same as in the case of the RAM Transfer command.

2. From the controller to the device

   The data in the 3rd byte is operation command data. The Product Information Read command data (30H) is sent here.

3. From the device to the controller

   The data in the 4th byte is the ACK response data to the operation command data in the 3rd byte.
   The device first checks to see if the data in the 3rd byte contains any error. If a receive error is found, the device returns the ACK response data for communications error (bit 3) x8H and waits for the next operation command data (3rd byte). The upper four bits of the ACK response data are undefined. (They are the upper four bits of the immediately preceding operation command data.)
   Then, if the data in the 3rd byte corresponds to one of the operation command values given in Table 3.2.7, the device echoes back the received data (ACK response for normal reception). In this case, 30H is returned and execution then branches to the product information read processing routine. If the data in the 3rd byte does not correspond to any operation command, the device returns the ACK response data for operation command error (bit 0) x1H and waits for the next operation command data (3rd byte). The upper four bits of the ACK response data are undefined. (They are the upper four bits of the immediately preceding operation command data.)

4. From the device to the controller

   The data in the 5th to 8th bytes is the data stored at addresses 02FEF0H to 02FEF3H in the flash memory. By writing the ID information of software at these addresses, the version of the software can be managed. (For example, 0002H can indicate that the software is now in version 2.)

5. From the device to the controller

   The data in the 9th to 20th bytes denotes the part number of the device. 'TMP91FW27_ _ _' is sent in ASCII code starting from the 9th byte.
   Note: An underscore ('_') indicates a space.

6. From the device to the controller

   The data in the 21st to 24th bytes is the password comparison start address. F4H, FEH, 02H and 00H are sent starting from the 21st byte.

7. From the device to the controller

   The data in the 25th to 28th bytes is the RAM start address. 00H, 10H, 00H and 00H are sent starting from the 25th byte.

8. From the device to the controller

   The data in the 29th to 32nd bytes is the RAM (user area) end address. FFH, 3DH, 00H and 00H are sent starting from the 29th byte.

Not Recommended for New Design

#### 3.2.4.15 Password

When the RAM Transfer command (10H) or the Flash Memory Protect Set command (60H) is received as operation command data, password verification is performed. First, the device echoes back the operation command data (10H to 60H) and checks the data (12 bytes) in the password area (addresses 02FEF4H to 02FEFFH).

Then, the device verifies the password data received in the 5th to 16th bytes against the data in the password area as shown in Table 3.2.20.

Unless all the 12 bytes are verified correctly, a password error will occur.

A password error will also occur if all the 12 bytes of password data contain the same value. Only exception is when all the 12 bytes are "FFH" and verified correctly and the reset vector area (addresses 02FF00H to 02FF02H) is all "FFH". In this case, a blank device will be assumed and no password error will occur.

If a password error has occurred, the device returns the ACK response data for password error in the 18th byte.

Table 3.2.20 Password Verification Table

| Receive data | Data to be verified against |
|---|---|
| 5th byte | Data at address 02FEF4H |
| 6th byte | Data at address 02FEF5H |
| 7th byte | Data at address 02FEF6H |
| 8th byte | Data at address 02FEF7H |
| 9th byte | Data at address 02FEF8H |
| 10th byte | Data at address 02FEF9H |
| 11th byte | Data at address 02FEFAH |
| 12th byte | Data at address 02FEFBH |
| 13th byte | Data at address 02FEFCH |
| 14th byte | Data at address 02FEFDH |
| 15th byte | Data at address02FEFEH |
| 16th byte | Data at address 02FEFFH |

Example of data that cannot be specified as a password

For blank products (Note)

・The password of a blank product must be all "FFH" (FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH).

Note: A blank product is a product in which all the bytes in the password area (addresses 02FEF4H to 02FEFFH) and the reset vector area (addresses 02FF00H to 02FF02H) are "FFH".

For programmed products

・The same 12 consecutive bytes cannot be specified as a password.

The table below shows password error examples.

| Programmed product | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Note |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Error example 1 | FFH | FFH | FFH | FFH | FFH | FFH | FFH | FFH | FFH | FFH | FFH | FFH | All "FF" |
| Error example 2 | 00H | 00H | 00H | 00H | 00H | 00H | 00H | 00H | 00H | 00H | 00H | 00H | All "00" |
| Error example 3 | 5AH | 5AH | 5AH | 5AH | 5AH | 5AH | 5AH | 5AH | 5AH | 5AH | 5AH | 5AH | All "5A" |

Not Recommended for New Design

### 3.2.5 User Boot Mode (in Single Chip Mode)

User Boot mode, which is a sub mode of Single Chip mode, enables a user-created flash memory program/erase routine to be used. To do so, the operation mode of Single Chip mode must be changed from Normal mode for executing a user application program to User Boot mode for programming/erasing the flash memory.

For example, the reset processing routine of a user application program may include a routine for selecting Normal mode or User Boot mode upon entering Single Chip mode. Any mode-selecting condition may be set using the device's I/O to suit the user system.

To program/erase the flash memory in User Boot mode, a program/erase routine must be incorporated in the user application program in advance. Since the processor cannot read data from the internal flash memory while it is being programmed or erased, the program/erase routine must be executed from the outside of the flash memory. While the flash memory is being programmed/erased in User Boot mode, interrupts must be disabled.

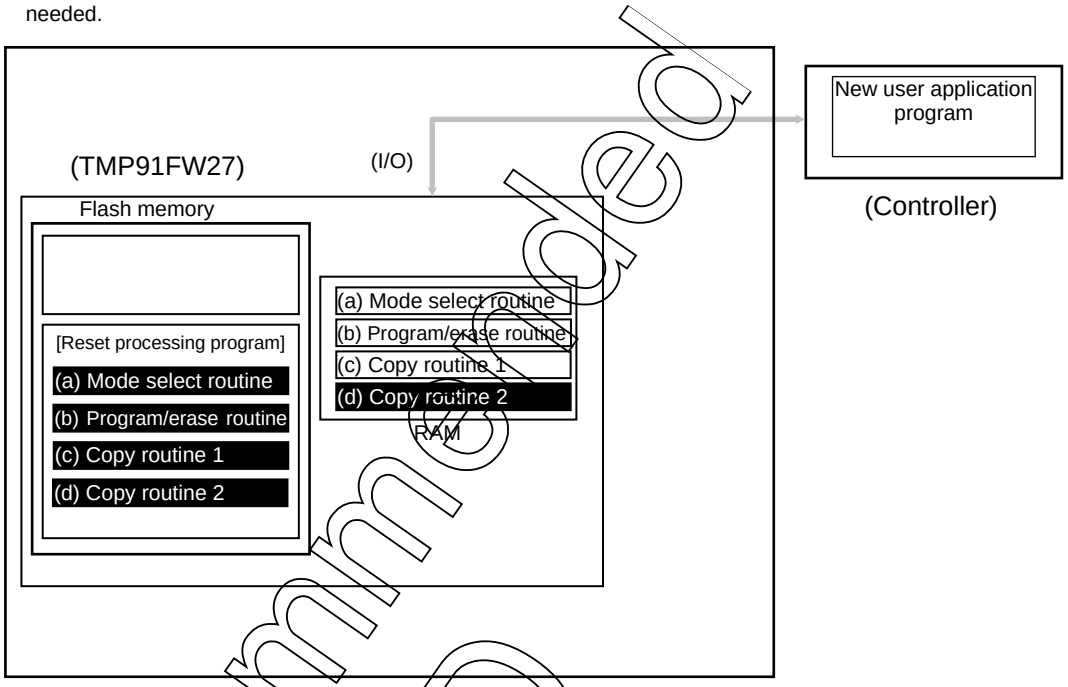
The pages that follow explain the procedure for programming the flash memory using two example cases. In one case the program/erase routine is stored in the internal flash memory (1-A); in the other the program/erase routine is transferred from an external source (1-B).

Not Recommended for New Design

*(Step-5) Restoring the user boot program in the flash memory*

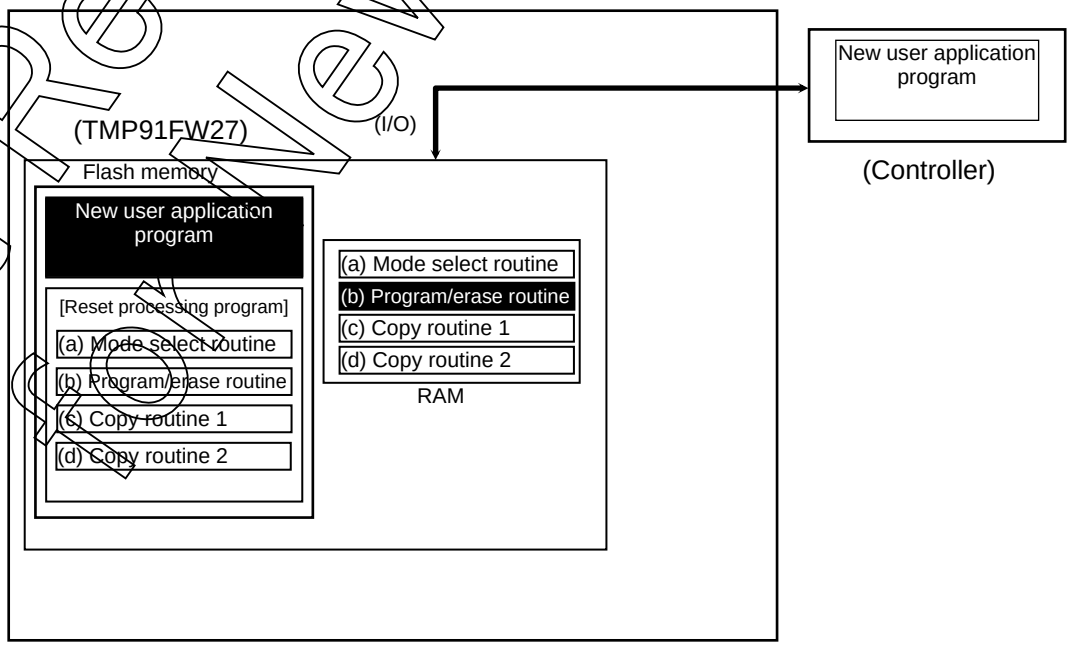
The copy routine 2 (d) in the RAM copies the routines (a) to (d) into the flash memory.

Note: If data is erased on a sector basis and the routines (a) to (d) are left in the flash memory, step 5 is not needed.




*(Step-6) Writing the new user application program to the flash memory*

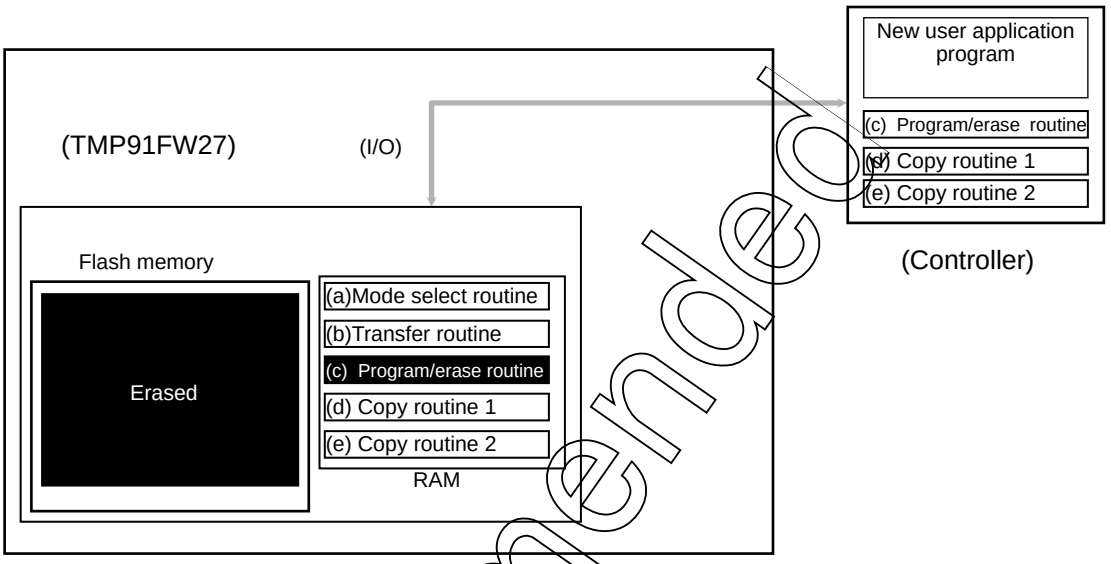
The program/erase routine in the RAM is executed to load the new user application program from the controller into the erased area of the flash memory.




Not Recommended for New Design

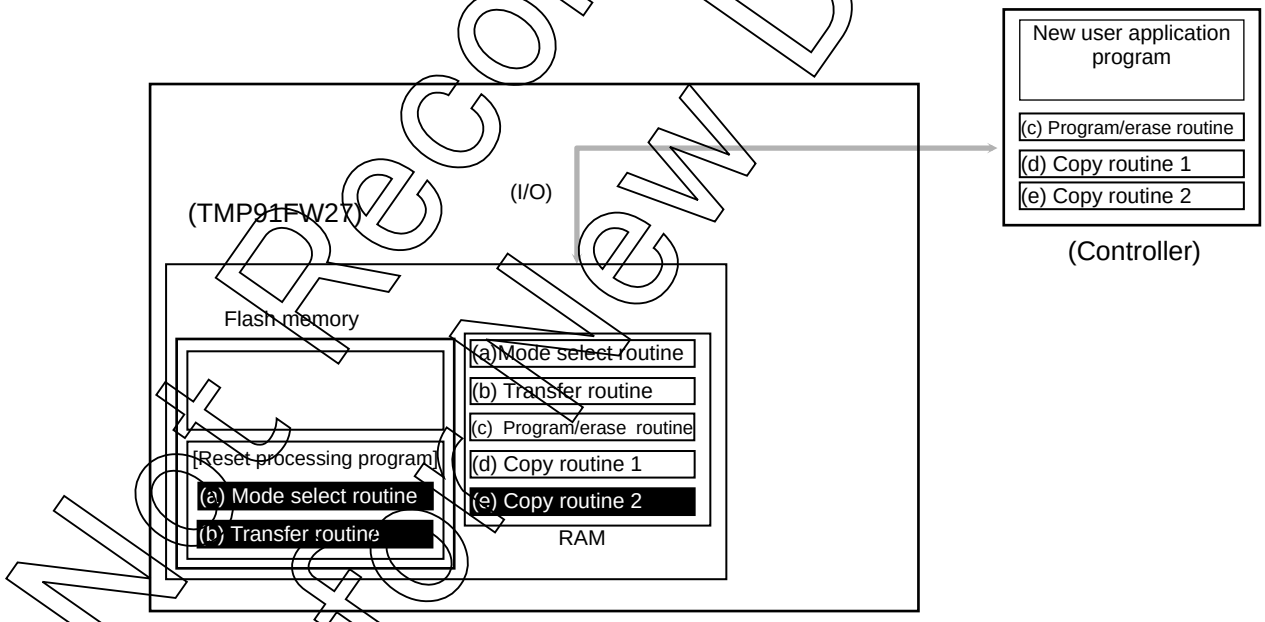
*(Step-5) Erasing the flash memory by the program/erase routine*

The program/erase routine (c) erases the old user program area.




*(Step-6) Restoring the user boot program in the flash memory*

The copy routine (e) copies the routines (a) and (b) from the internal RAM into the flash memory.




Not Recommended for New Design

2) Toggle bit (D6)

When the Flash Memory Program, Sector Erase, Chip Erase, Write Protect Set, or Read Protect Set command sequence is executed, bit 6 (D6) of the data read by read operations outputs “0” and “1” alternately each time it is read until the processing of the executed command sequence has completed. The toggle bit (D6) thus provides a software means of checking whether or not the processing of each command sequence has completed. Normally, the same address in the flash memory is read repeatedly until the same data is read successively. The initial read of the toggle bit always returns “1”.

Note: The flash memory incorporated in the TMP91FW27 does not have an exceed-time-limit bit (D5). It is therefore necessary to set the data polling time limit and toggle bit polling time limit so that polling can be stopped if the time limit is exceeded.

#### 3.2.6.9 Data Read

Data is read from the flash memory in byte units or word units. It is not necessary to execute a command sequence to read data from the flash memory.

Not Recommended for New Design

#### 3.2.6.10 Programming the Flash Memory by the Internal CPU

The internal CPU programs the flash memory by using the command sequences and hardware sequence flags described above. However, since the flash memory cannot be read during auto operation mode, the program/erase routine must be executed outside of the flash memory.

The CPU can program the flash memory either by using Single Boot mode or by using a user-created protocol in Single Chip mode (User Boot).

1) Single Boot:

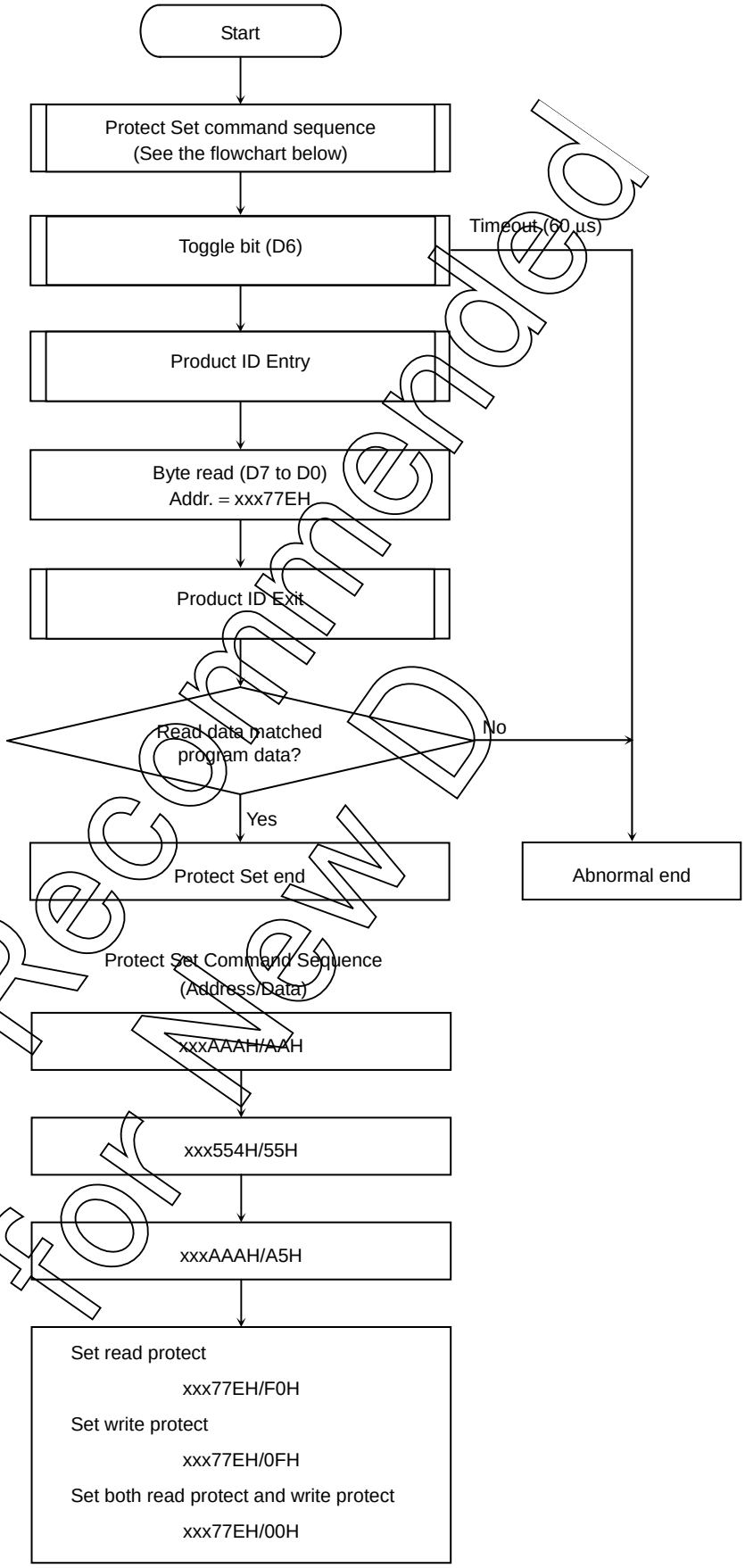
The microcontroller is started up in Single Boot mode to program the flash memory by the internal boot ROM program. In this mode, the internal boot ROM is mapped to an area including the interrupt vector table, in which the boot ROM program is executed. The flash memory is mapped to an address area different from the boot ROM area. The boot ROM program loads data into the flash memory by serial transfer. In Single Boot mode, interrupts must be disabled including non-maskable interrupts ( $\overline{\text{NMI}}$ , etc.).
For details, see 3.2.4"Single Boot Mode".

2) User Boot:

In this method, the flash memory is programmed by executing a user-created routine in Single Chip mode (normal operation mode). In this mode, the user-created program/erase routine must also be executed outside of the flash memory. It is also necessary to disable interrupts including non-maskable interrupts.
The user should prepare a flash memory program/erase routine (including routines for loading write data and writing the loaded data into the flash memory). In the main program, normal operation is switched to flash memory programming operation to execute the flash memory program/erase routine outside of the flash memory area. For example, the flash memory program/erase routine may be transferred from the flash memory to the internal RAM and executed there or it may be prepared and executed in external memory.
For details, see 3.2.5"Flash Memory".

Not Recommended for New Design

Read/Write Protect Set




Not Recommended for New Design

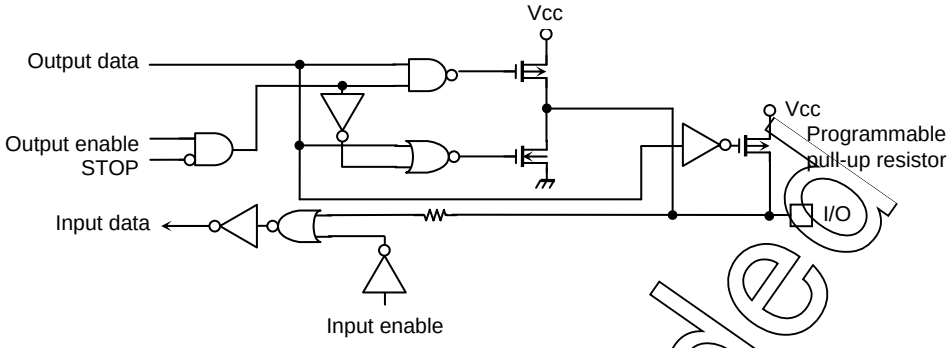
## 4.2 DC Characteristics (1/2)

| Parameter | | Symbol | Condition | | Min | Typ.(Note) | Max | Unit |
|---|---|---|---|---|---|---|---|---|
| Power supply voltage (AVCC = DVCC, AVSS = DVSS = 0 V) | | VCC | fc = 4 to 27 MHz | fs = 30 to 34 kHz | 2.7 | | 3.6 | V |
| | | | fc = 2 to 16 MHz | | 2.2 | | | |
| Power supply voltage (AVCC = DVCC, AVSS = DVSS = 0 V) for erase/program operations of flash memory | | VCC | fc = 4 to 27 MHz, Ta = −10~40°C | | 2.7 | | 3.6 | V |
| Input low voltage | P00 to P17 (AD0 to AD15) | VIL | Vcc ≥ 2.7 V | | −0.3 | | 0.6 | V |
| | | | Vcc < 2.7 V | | | | 0.2 Vcc | |
| | P20 to P97 (Except P63) | VIL1 | Vcc ≥ 2.7 V | | | | 0.3 Vcc | |
| | | | Vcc < 2.7 V | | | | 0.2 Vcc | |
| | $\overline{\text{RESET}}$ , $\overline{\text{NMI}}$ P63 (INT0) | VIL2 | Vcc ≥ 2.7 V | | | | 0.25 Vcc | |
| | | | Vcc < 2.7 V | | | | 0.15 Vcc | |
| | AM0 and AM1 | VIL3 | Vcc ≥ 2.7 V | | | | 0.3 | |
| | | | Vcc < 2.7 V | | | | 0.3 | |
| | X1 | VIL4 | Vcc ≥ 2.7 V | | | | 0.2 Vcc | |
| | | | Vcc < 2.7 V | | | | 0.1 Vcc | |
| Input high voltage | P00 to P17 (AD0 to AD15) | VIH | Vcc ≥ 2.7 V | | 2.0 | | Vcc + 0.3 | V |
| | | | Vcc < 2.7 V | | 0.7 Vcc | | | |
| | P20 to P97 (Except P63) | VIH1 | Vcc ≥ 2.7 V | | 0.7 Vcc | | | |
| | | | Vcc < 2.7 V | | 0.8 Vcc | | | |
| | $\overline{\text{RESET}}$ , $\overline{\text{NMI}}$ , P63 (INT0) | VIH2 | Vcc ≥ 2.7 V | | 0.75 Vcc | | | |
| | | | Vcc < 2.7 V | | 0.85 Vcc | | | |
| | AM0 and AM1 | VIH3 | Vcc ≥ 2.7 V | | Vcc − 0.3 | | | |
| | | | Vcc < 2.7 V | | Vcc − 0.3 | | | |
| | X1 | VIH4 | Vcc ≥ 2.7 V | | 0.8 Vcc | | | |
| | | | Vcc < 2.7 V | | 0.9 Vcc | | | |
| Output low voltage | | VOL | IOL = 1.6mA | Vcc ≥ 2.7 V | | | 0.45 | V |
| | | | IOL = 0.4mA | Vcc < 2.7 V | | | 0.15 Vcc | |
| Output high voltage | | VOH | IOH = −400 μA | Vcc ≥ 2.7 V | Vcc − 0.3 | | | |
| | | | IOH = −200 μA | Vcc < 2.7 V | 0.8 Vcc | | | |

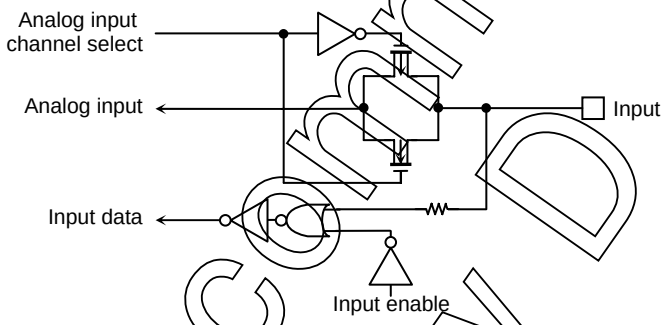
Note: Typical values are for when Ta = 25°C and $V_{CC}$ = 3.0 V uncles otherwise noted.

Not Recommended for New Design

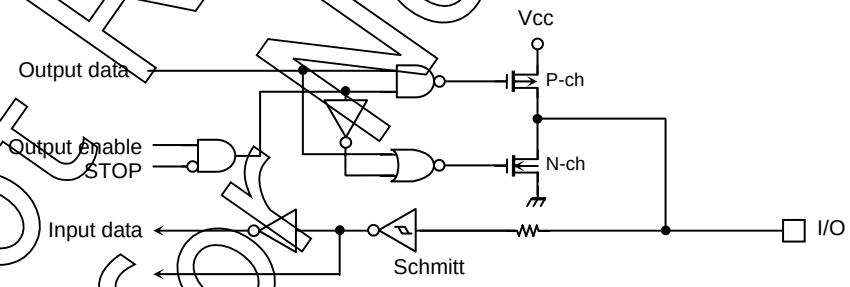
P32, P40~P42




■ P5 (AN0~AN3)

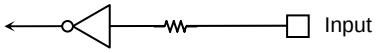



■ P63 (INT0)

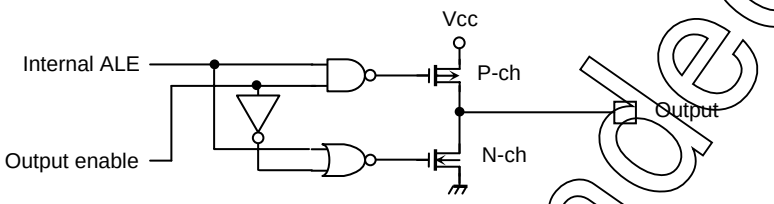



Not Recommended for New Design

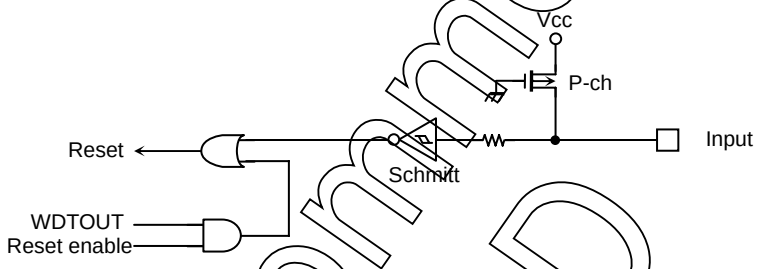
■ AM0~AM1



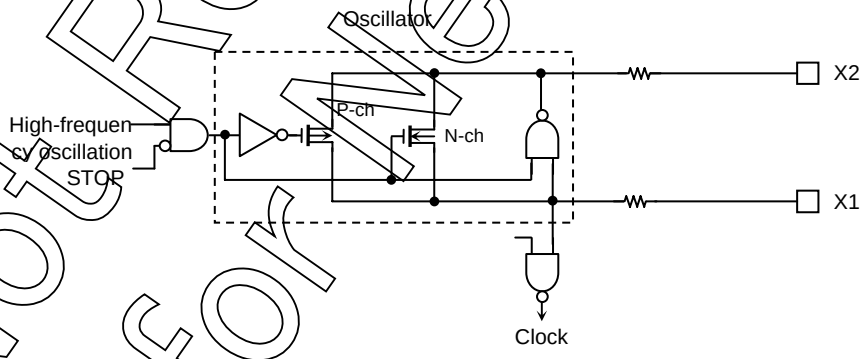

■ ALE




■ $\overline{\text{RESET}}$



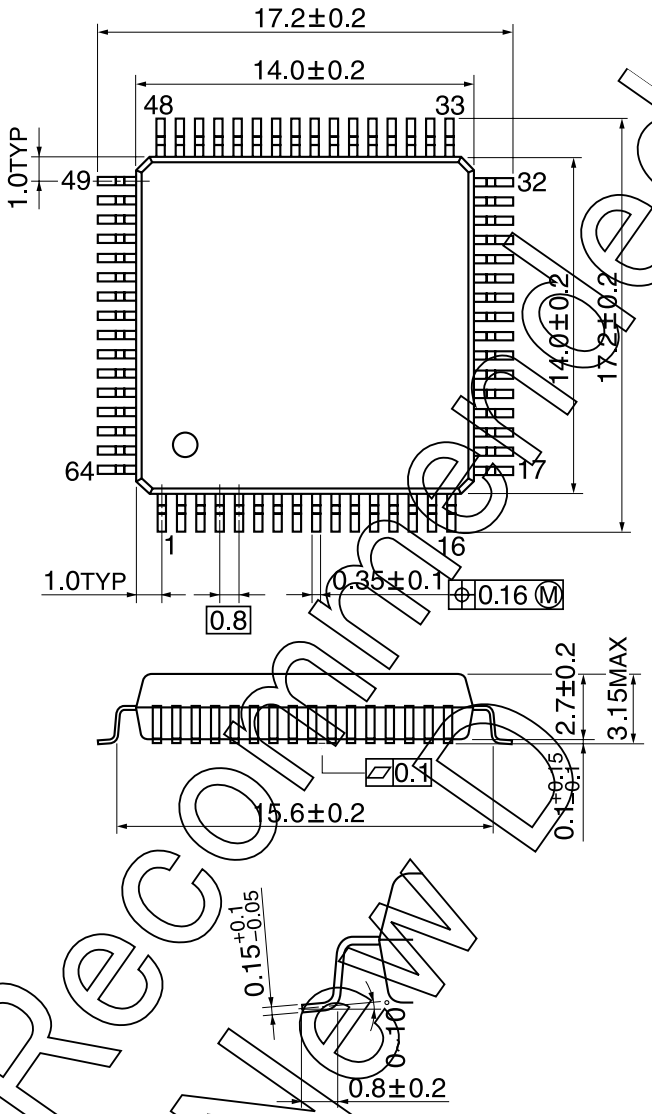

■ X1, X2




Not Recommended for New Design

QFP64-P-1414-0.80A

Unit: mm




Not Recommended for New Design