FEXL66Q591-01

OKI

MSM66Q591
FLASH MEMORY

User's Manual

First Edition October 2000

Oki Electric Industry Co., Ltd.

CONTENTS

1. OVERVIEW

1.1 Overview

The MSM66Q591 is the MSM66591 with the internal program memory (128K byte mask ROM) replaced by an electrically erasable and programmable non-volatile memory (128K byte flash memory). The MSM66Q591 makes it possible to rewrite the program before and after installing in the system using three types of flash memory programming modes.

1.2 Features

The flash memory of the MSM66Q591 can be programmed using only a single 5 V power supply. Further, in order to prevent the data being destroyed due to erroneous operation of the flash memory, there is a power supply voltage sensing circuit that prohibits programming of the flash memory when the power supply voltage falls below about 3 V. In addition, this LSI has a built-in power ON reset circuit that prohibits programming for about 12 ms after the power supply is switched ON.
The features of the flash memory in the MSM66Q591 are the following.

- Programming modes
  The following three types of programming modes are available for the flash memory of the MSM66Q591.
  - Parallel mode: For pre-shipment testing of the microcomputer.
  - Serial mode: Programming using a dedicated or a general-purpose serial programmer.
  - User mode: Programming based on program execution.

- Auto-erase function
  Since this LSI is provided with the auto-erase function by which the sector to be programmed is automatically erased before programming the flash memory, there is no need to carry out separately an erasing operation before programming.

- Programming time
  The time taken for programming the flash memory is as follows.
  - Sector (128 bytes) programming: 40 ms (max)
  - Sector (128 bytes) programming (with auto-erase): 45 ms (max)
  - Setting the security bit: 400 ms (max)

- Erasing time
  - 128K bytes entire chip erasing: 10 ms (max)
  - 128K bytes entire chip erasing (including the security bit): 410 ms (max)
  - Note: This is dedicated to the user mode.

- Security function
  The flash memory is incorporated with the security function that prohibits reading out or programming the contents of the memory from an external device. The security function is effective for the 128K byte flash memory (internal ROM) and the 4K byte flash memory (CROM). The setting of the security function can be made by setting or resetting the security bit. When the security bit is set ("1"), the security function becomes effective, and it will not be possible to program the contents of the flash memory from an external device in any programming mode. Also, when the flash memory is read out in this condition, the data "00H" is output irrespective of the actual contents of the flash memory (in both the serial and parallel mode). When the security bit is reset ("0"), the security function is disabled, and the flash memory contents can be read out or rewritten in each of the programming modes. Further, in the user mode, the resetting of the security bit is done automatically when the entire chip is erased.
  Carry out the setting of the security function in the serial mode or in the user mode.

2. PROGRAMMING MODES

2.1 Overview of the Programming Modes

The MSM66Q591 has the following three programming modes.

(1) Parallel mode
This is the mode in which the programming is done by the microcomputer alone. The programming is done before being installed in the user's application system. The parallel mode is used mainly for pre-shipment testing of the microcomputer.

(2) User mode
This is the mode used for programming with the chip installed in the system. It is possible to program the flash memory after the chip has been installed in the user's application system.
In the user mode, the programming is done by operating the CPU. The write program will have to be prepared by the users themselves.

(3) Serial mode
This is the mode in which the programming is done with the chip installed in the system similar to the user mode.
In the serial mode, the programming is done when the microcomputer is either in the reset state or in the stop mode. This programming is done using either the OKI Electric Flash Memory Dedicated Programmer (FW66500S) or the general purpose flash memory microcomputer programmer (AF200) manufactured by Yokogawa Digital Computers.

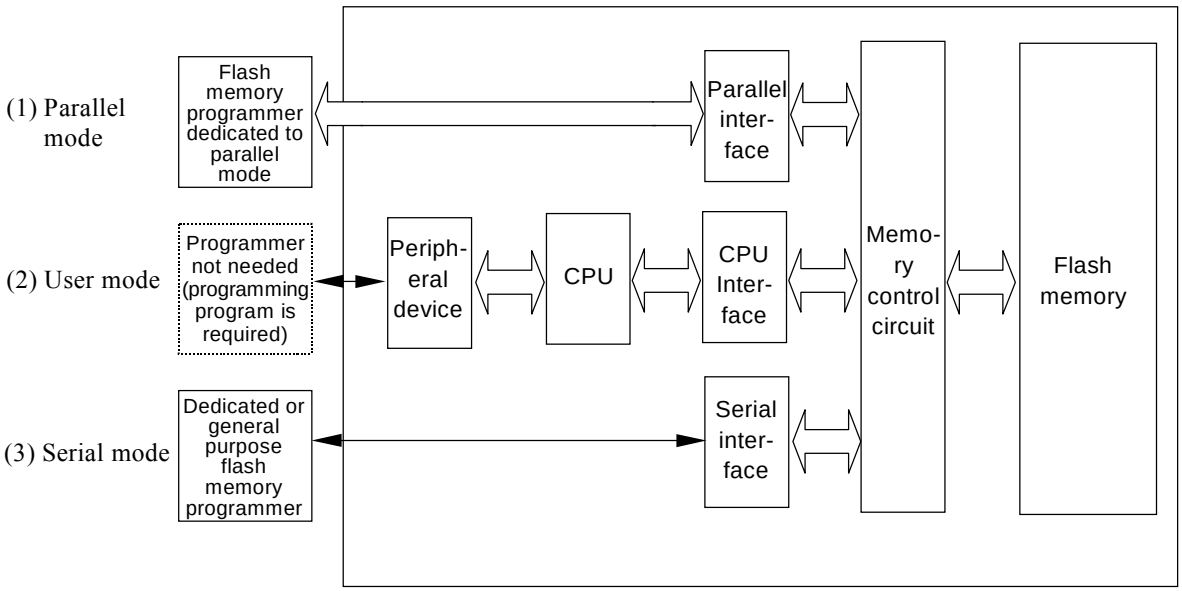


Figure 2-1 Schematic block diagram of programming modes

The schematic block diagram of the flash memory programming modes is shown in Figure 2-1.

2.2 Parallel Mode

This is the mode in which the programming is done by the microcomputer alone.
The parallel mode is used mainly for pre-shipment testing of the microcomputer. The flash memory programming is done using a flash memory programmer dedicated to the parallel mode.
Further, it is not possible to set or reset the security bit in the parallel mode. Carry out the setting or resetting of the security bit in the serial mode or in the user mode.

2.3 User Mode

2.3.1 Overview of user mode

The user mode is one in which the user programs the flash memory when it is already installed in the system. The programming is possible even after the microcomputer is installed in the PCB of the user's application system.
The MSM66Q591 has a built-in 128K byte and a 4K byte flash memory. The 128K byte flash memory is used as the internal program memory (internal ROM). The 4K byte flash memory is used as the control program memory (CROM) for programming the internal ROM in the user mode.
When set to the user mode, the program in the CROM becomes effective. A program specifying the data I/O method for programming the internal program memory should be stored beforehand in the CROM. The CPU executes the program in the CROM, and stores 128 bytes of data for programming the internal ROM in the addresses 300H to 37FH of the RAM using the peripheral device. Thereafter, by executing programming under SFR control dedicated to the user mode, the contents of the addresses 300H to 37FH of the RAM (128 bytes) are transferred to the internal ROM. Therefore, in the user mode, by changing the program in the CROM, it is possible to set freely the I/O method of the programming data using the peripheral device.
Further, for setting the data in the CROM, it is necessary to write beforehand the program in either the parallel mode or in the serial mode. Also, the program in the CROM can only be executed in the user mode.
The schematic block diagram of the user mode is shown in Figure 2-2.

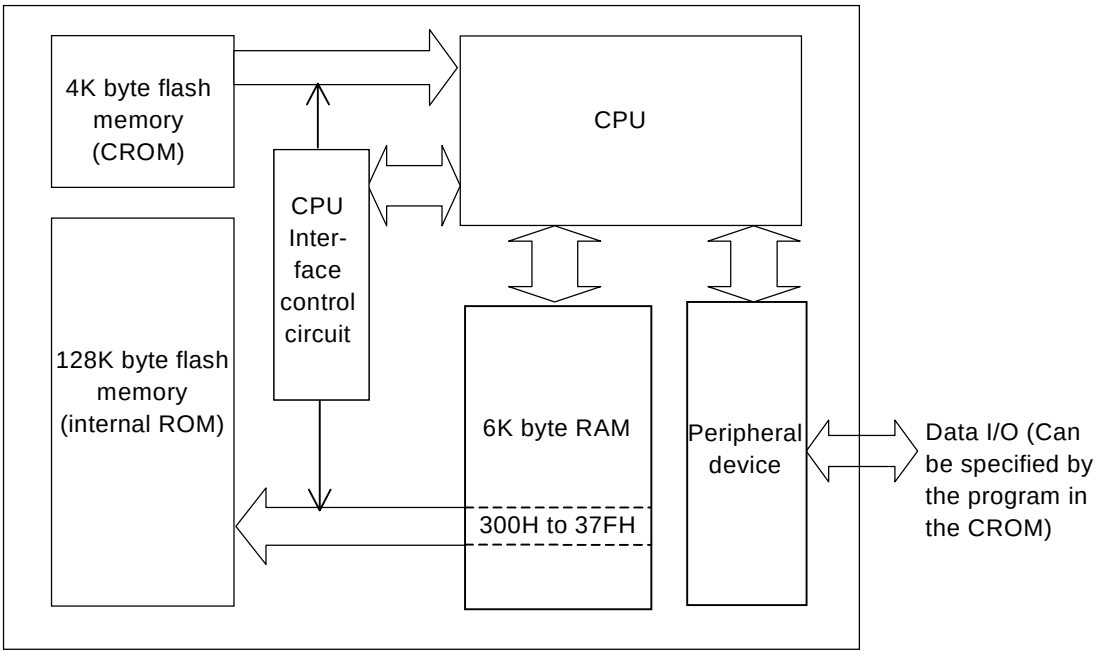


Figure 2-2 Schematic block diagram of user mode

2.3.2 Setting the user mode

The user mode can be set by releasing the reset state (changing the $\overline{RES}$ pin input from the L level to the H level) with H level signals being applied to the $\overline{EA}$ and TEST pins. When set to the user mode, the CPU executes the program in the CROM. Further, since the TEST pin is sampled by the reset signal based on the input to the $\overline{RES}$ pin, keep the TEST pin tied to the H level for 100 ns or more before and after the reset state is released.

The user mode is released by carrying out a reset (changing the $\overline{RES}$ pin input from the H level to the L level) with an L level signal being applied to the TEST pin. Even in this case, since the TEST pin is sampled by the reset signal based on the input to the $\overline{RES}$ pin, keep the TEST pin tied to the L level for 100 ns or more before and after the reset state is initiated. Thereafter, when the reset state is released (the $\overline{RES}$ pin input is taken from the L level to the H level), the CPU executes the program in the internal ROM.

Note that, even after the programming or reading is completed in the user mode, the completion of the programming or reading out from an external device is not assured. Therefore, set a program in the CROM that posts the completion of programming or reading out to the external device after such a processing is completed, carry out a reset from the external device after posting (changing the $\overline{RES}$ pin input from the H level to the L level), thereby releasing the user mode.

The setting pins for the user mode are shown in Table 2-1, and the timing chart of user mode setting and releasing is shown in Figure 2-3.

Table 2-1 User mode setting pins

Pin No.	Primary function	User mode setting condition
3	$\overline{EA}$	H Level
92	TEST	H Level
2	$\overline{RES}$	Reset state is released

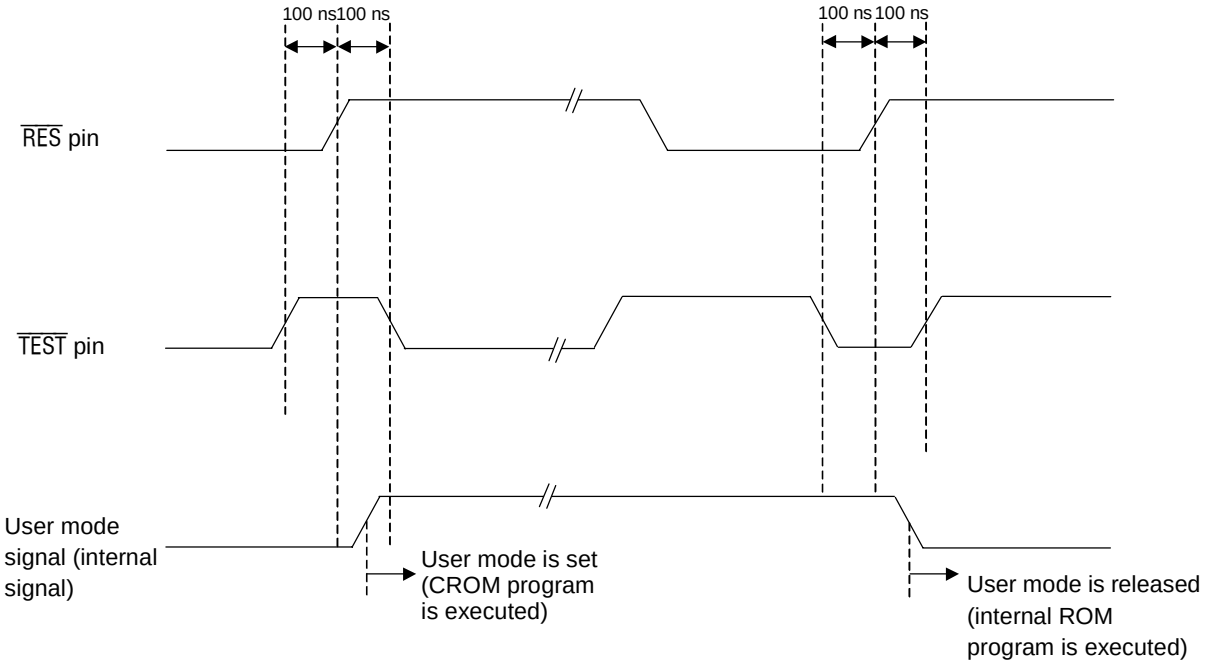


Figure 2-3 Timing chart of setting and releasing the user mode.

2.3.3 Programming in the user mode

The MSM66Q591 has a built-in SFR that can only be accessed during the user mode for programming the flash memory in the user mode.

The programming in the user mode is done by controlling the flash memory control register (FLACON), the flash memory address register (FLAADRS), the flash memory acceptor register (FLAACP), and the flash memory output buffer register (FLABUF), which are the special function registers (SFRs) dedicated to the user mode.

Table 2-2 gives a list of the special function registers dedicated to the user mode.

In all modes other than the user mode, writing to the registers FLACON, FLAADRS, FLAACP, and FLABUF is disabled, and if they are read out, always the value 0FFH or 0FFFFH is output irrespective of the content of the register.

Table 2-2 List of SFRs dedicated to the user mode

Address [H]	Name	Abbreviation (byte)	Abbreviation (Word)	R/W	8/16 operation	Reset state
01F0*	Flash memory acceptor	FLAACP	—	W	8	"0"
01F1*	Flash memory control register	FLACON	—	R/W	8	20
01F2*	Flash memory address register	—	FLAADRS	R/W	16	Uncertain
01F3*						
01F4	Flash memory output buffer register	FLABUF	—	R	8	Uncertain

Notes: 1. In modes other than the user mode, writing to the SFRs dedicated to the user mode will not be valid, and if they are read out, all bits will be read out as "1".
2. The addresses with an asterisk (*) are those with missing bits.

The watchdog timer (WDT) will not stop during the hold mode for programming the flash memory, so do not activate the watchdog timer (WDT).

(1) Flash memory address register (FLAADRS)
FLAADRS is a register for setting the programming address in bits 10 to 1 (FA16 to 7) when programming the flash memory (internal ROM).
The configuration of FLAADRS is shown in Figure 2-4.

FLAADRS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	TLPC	FA16	FA15	FA14	FA13	FA12	FA11	FA10	FA9	FA8	FA7	—

During programming

FA16 to 7										Programming address
16	15	14	13	12	11	10	9	8	7	
0	0	0	0	0	0	0	0	0	0	00000H to 0007FH
0	0	0	0	0	0	0	0	0	1	00080H to 000FFH
1	1	1	1	1	1	1	1	1	0	1FF00H to 1FF7FH
1	1	1	1	1	1	1	1	1	1	1FF80H to 1FFFFH

During erasing the entire chip

FA16 to 7										Entire chip erasing address
16	15	14	13	12	11	10	9	8	7	
*	*	*	*	*	*	*	*	*	*	00000H to 1FFFFH

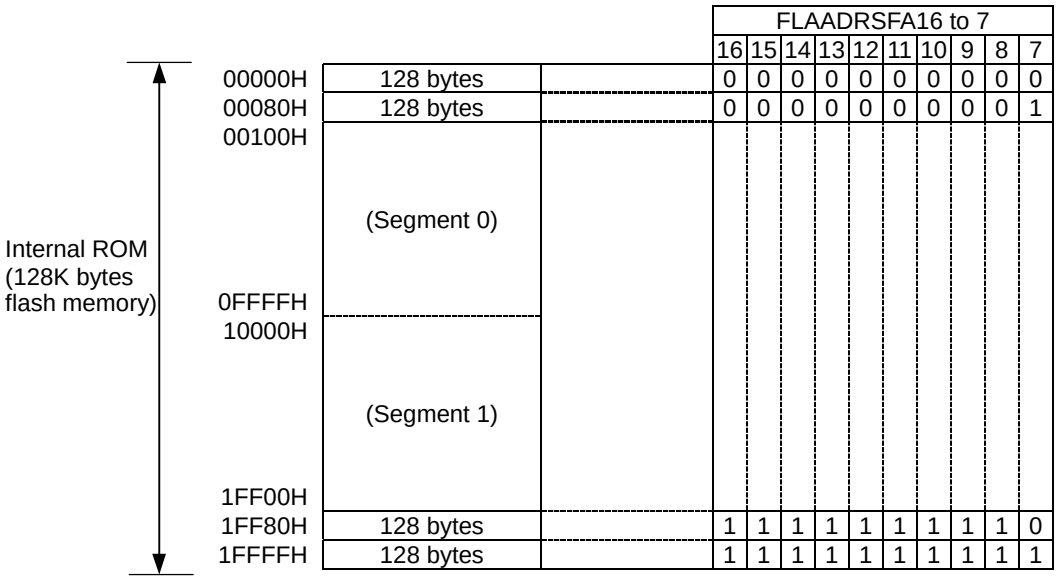
Note: An asterisk (*) indicates either "0" or "1".

0	When the internal ROM is the target ROM of the ROM table reference instruction
1	When the CROM is the target ROM of the ROM table reference instruction

A hyphen (—) indicates a non-existent bit and returns a "1" when read out.

Figure 2-4 Configuration of FLAADRS

The internal ROM of the MSM66Q591 is configured by segment 0 and segment 1 of 64K bytes each, and the address specified by FLAADRS is a 128K-byte space formed by connecting segment 0 and segment 1 in series. However, since FLAADRS permit only 16-bit operations, it is not possible to use the bit manipulation instructions such as SB, RB, etc. The content of FLAADRS will be indefinite immediately after carrying out a reset.
The relationship among segment 0, segment 1, and FLAADRS (the FLAADRS space) is shown in Figure 2-5.



			FLAADRSFA16 to 7									
			16	15	14	13	12	11	10	9	8	7
00000H	128 bytes		0	0	0	0	0	0	0	0	0	0
00080H	128 bytes		0	0	0	0	0	0	0	0	0	1
00100H	(Segment 0)											
0FFFFH												
10000H	(Segment 1)											
1FF00H												
1FF80H	128 bytes		1	1	1	1	1	1	1	1	1	0
1FFFFH	128 bytes		1	1	1	1	1	1	1	1	1	1

Figure 2-5 The FLAADRS memory map

- FA16 to FA7 (Bit 10 to Bit 1)
  Bits 10 to 1 of FLAADRS (FA16 to 7) indicate the higher order 10 bits of the programming address in the 128K byte space, and are used to set the address area in units of 128 bytes. Also, there is no need to set the address in FLAADRS in the case of an entire chip erasure.
  - Writing (128 bytes): 10 bits (FA16 to 7) are valid.
  - Entire chip erasure (128K bytes): Address setting is not required.

- TLPC (Bit 11)
  The verification of the data written into the flash memory is done using ROM table reference instructions.
  The bit 11 (TLPC) of FLAADRS is the bit for selecting either the internal ROM or the CROM as the target ROM for the ROM table reference instruction. When executing a ROM table reference instruction, always write a "0" or a "1" in bit 11 (TLPC) of FLAADRS thereby specifying the target ROM.

  - When TLPC is "0", the internal ROM becomes the target of the ROM table reference instruction. (Use this function when wanting to reference the data in the internal ROM.)

  Example 1: When the target of access is Segment 0 of the internal ROM.

```
MOVB MEMSACP,#50H:
MOVB MEMSACP,#0A0H:
SB LROM:                Extends the internal ROM to 128K bytes.
MOVB TSR,#00H:          Specifies segment 0.
MOV ALRB,#0002H:        The local register is set to 210H to 217H.
MOV FLAADRS,#0000H:     Bit 11 (TLPC) of FLAADRS is set to "0".
                        (FA16 to 7 are immaterial in a ROM table reference
                        instruction.)
MOV ER0,#1234H:
LCB A,[ER0]:            The ROM data at the address 1234H of segment 0 of the
                        internal ROM is loaded in ACC.
```

Example 2: When the target of access is Segment 1 of the internal ROM.

```
MOVB MEMSACP,#50H:
MOVB MEMSACP,#0A0H:
SB LROM:                 Extends the internal ROM to 128K bytes.
MOVB TSR,#01H:           Specifies segment 1.
MOV ALRB,#0002H:         The local register is set to 210H to 217H.
MOV FLAADRS,#0000H:      Bit 11 (TLPC) of FLAADRS is set to "0".
                         (FA16 to 7 are immaterial in a ROM table reference
                         instruction.)
MOV ER0,#1234H:
LCB A,[ER0]:             The ROM data at the address 1234H of segment 1 of the
                         internal ROM is loaded in ACC.
```

- When TLPC is set to "1", the CROM itself becomes the target. (Use this function when accessing the data of the CROM. Also, in the case of the CROM, always specify segment 0.)

Example 3:

```
MOVB MEMSACP,#50H:
MOVB MEMSACP,#0A0H:
SB LROM:
MOVB TSR,#00H:           In the case of the CROM, always specify segment 0.
MOV ALRB,#0002H:         The local register is set to 210H to 217H.
MOV FLAADRS,#0800H:      Bit 11 (TLPC) of FLAADRS is set to "1".
                         (FA16 to 7 are immaterial in a ROM table reference
                         instruction.)
MOV ER0,#0123H:
LCB A,[ER0]:             The ROM data at the address 1234H of the CROM is loaded
                         in ACC.
```

(2) Flash memory acceptor (FLAACP)

The FLAACP register is an acceptor register used when setting data in the flash memory control register (FLACON). This acceptor is set to "1" when n5H, nAH (N = 0 to F) are written successively, and writing into the flash memory control register (FLACON) becomes enabled. Further, this acceptor gets reset to "0" when a programming or an erasure is carried out on the internal ROM, and writing to FLACON becomes disabled.

The configuration of FLAACP is shown in Figure 2-6.

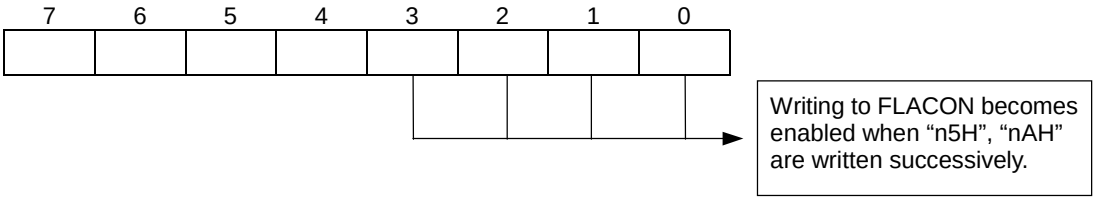
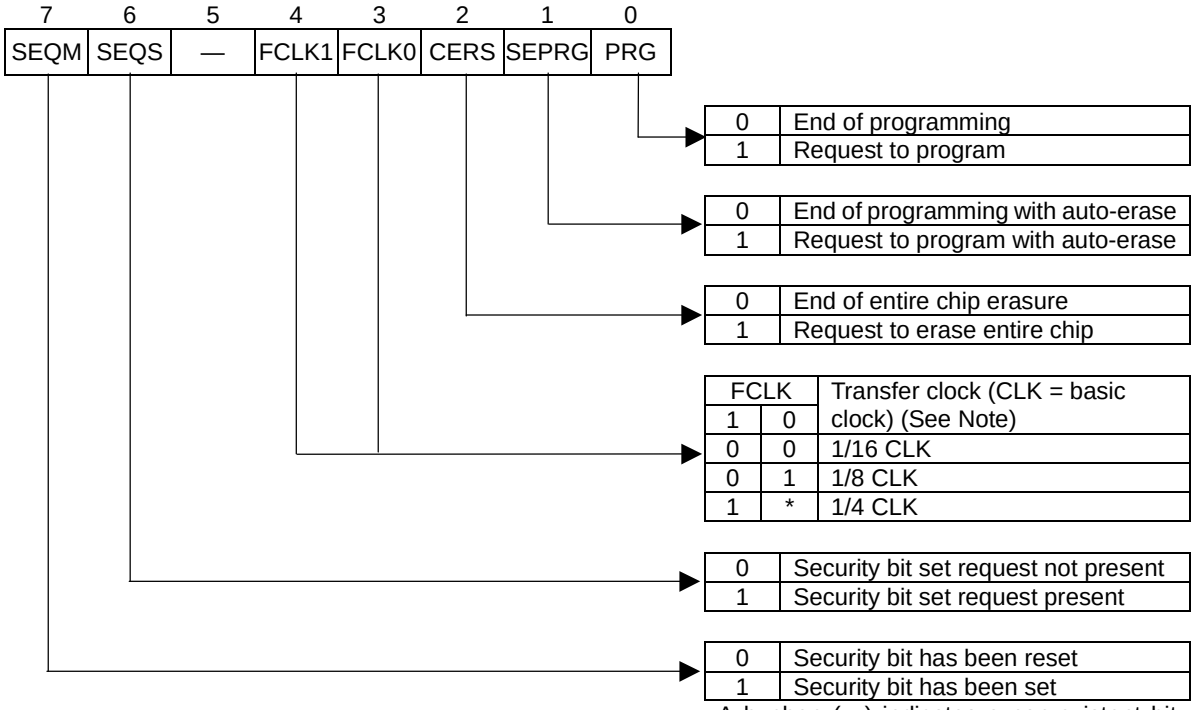


Figure 2-6 Configuration of FLAACP

(3) Flash memory control register (FLACON)
FLACON is a 7-bit register that controls flash memory programming and erasure.
The configuration of FLACON is shown in Figure 2-7.

FLACON



A hyphen (—) indicates a non-existent bit and returns a "1" when read out.
Note: An asterisk (*) indicates either "0" or "1".
Note: The basic clock (CLK) has twice the frequency of the source clock.

Figure 2-7 Configuration of FLACON

Writing into FLACON becomes enabled when the flash memory acceptor has been set (to "1") by successively writing n5H and nAH (N = 0 to F) in FLAACP. Also, FLAACP is reset (to "0") when a write or an erase operation is made on the internal ROM. When carrying out again a write or an erase operation on the flash memory, it is necessary to set (to "1") again the flash memory acceptor.

- PRG (Bit 0)
  This bit is used at the time of carrying out the first programming after the completion of erasing the entire chip. The programming is done by setting (to "1") the PRG flag (bit 0) of FLACON. When the PRG flag (bit 0) is set (to "1"), the CPU automatically goes into the hold state after executing one instruction, and the data at the addresses 300H to 37FH of the internal RAM is transferred to the internal ROM specified by FLAADRS thereby carrying out the programming operation. When the programming is completed, the hold state is released automatically, and also this bit will be reset (to "0"). For details of the hold state, see (5) "Hold mode" below.
  Further, it is necessary to set beforehand the address and data of writing in FLAADRS and the internal RAM. The relationship between the internal RAM and the internal ROM is shown in Figure 2-8.

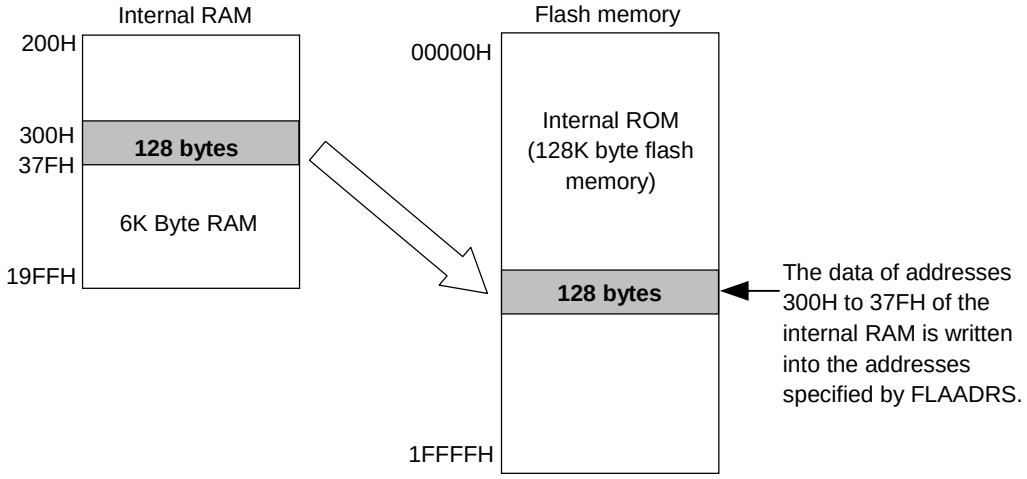


Figure 2-8 The relationship between the internal RAM and the internal ROM

- SEPRG (Bit 1)
  This bit is used when carrying out programming with the program written in the 128K byte internal ROM. In the case of this programming, it is not necessary to carry out the operation of erasing the entire chip because of the auto-erase function before programming. When carrying out the first programming after an entire chip erasure, use the PRG flag (bit 0).
  Programming with auto-erase is done by setting (to "1") the SEPRG flag (bit 1) of FLACON. When the SEPRG flag (bit 1) is set to "1", the CPU automatically goes to the hold state after executing one instruction, the 128 bytes of data specified by FLAADRS is erased, and the programming is done by transferring the data in the addresses 300H to 37FH of the internal RAM to the internal ROM. When the programming is completed, the hold state is automatically released and also this bit is reset to "0". For details of the hold state, see (5) "Hold mode" below.
  Further, it is necessary to set beforehand the address and data of writing in FLAADRS and the internal RAM.

- CERS (Bit 2)
  This bit is used when carrying out entire chip erasure of the 128K byte internal ROM and resetting of the security bit. Carrying out entire chip erasure of the 128K byte internal ROM and resetting of the security bit is done by setting (to "1") the CERS flag (bit 2) of FLACON. When the CERS flag (bit 2) is set (to "1"), the CPU automatically goes to the hold state after executing one instruction, and the entire chip erasure of the 128K byte internal ROM and resetting of the security bit are carried out. When the erasing is completed, the hold state is released automatically, and also this bit is reset (to "0"). For details of the hold state, see (5) "Hold mode" below. Further, it is not possible to carry out entire chip erasure and resetting of the securing bit independently of each other.
  When the data of the internal ROM is read out after carrying out an entire chip erasure, the data "0FFH" is returned over the entire 128K byte space.

- FCLK0, 1 (Bits 3, 4)
  The FCLK0 (bit 3) and FCLK1 (bit 4) flags of FLACON are the bits for setting the clock used at the time of transferring the data of the data in the addresses 300H to 37FH of the internal RAM to the internal ROM in the hold state. However, make sure to set these flags so that the transfer clock is 5 MHz or less.
  (Example: When 1/8CLK is selected, since the basic clock has twice the frequency of the source clock, and if the source clock frequency is 12 MHz, the transfer clock frequency will be 12 MHz × 2 × 1/8 = 3 MHz, and hence the transfer clock will be less than 5 MHz.)

- SEQS (Bit 6)
  This bit is a write-only flag that is used when setting the security bit of the 128K byte internal ROM and the 4K byte CROM. Setting of the security bit is made by setting (to "1") the SEQS flag (bit 6) of FLACON. When SEQS is set to "1", the CPU automatically goes to the hold state after executing one instruction, and the security bit will be set. When the setting of the security bit is completed, the hold state is released automatically, and also this bit is reset (to "0"). For details of the hold state, see (5) "Hold mode" below. Further, it is not possible to program the contents of the internal ROM when the security bit has been set. Also, read out the SEQM flag (bit 7) of FLACON in order to verify if the security bit has been set and the security function has been enabled. In addition, when reading out the SEQM flag immediately after the hold state is released, execute two NOP instructions before reading out the SEQM flag.

- SEQM (Bit 7)
  The SEQM flag (bit 7) of FLACON is the monitor flag for the security bit. The SEQM flag will be set (to "1") when the SEQS bit (bit 6) of FLACON has been set and the setting of the security bit has been completed. The SEQM flag will be reset (to "0") when the entire chip erasure operation and resetting of the security bit has been completed after the CERS bit (bit 2) of FLACON has been set (to "1"). This is a read-only flag. In addition, when reading out the SEQM flag immediately after the hold state is released, execute two NOP instructions before reading out the SEQM flag.
  When the SEQM flag has been set to "1", the security function will be enabled and it is not possible to program the contents of the internal ROM. However, even if the security function has been enabled, it is possible to read out the contents of the internal ROM using a ROM table reference instruction in the user mode.
  In order to confirm the contents of the internal ROM read out in the serial or parallel modes, execute successively a ROM table reference instruction followed by an instruction reading out the flash memory output buffer register (FLABUF).
  (For details, see (4) "Flash memory output buffer register (FLABUF)" below.) When the SEQM flag has been set (to "1"), the data "00H" will be read out from FLABUF irrespective of the contents of the internal ROM, and hence it is possible to verify that the security function has been enabled.

(4) Flash memory output buffer register (FLABUF)

FLABUF is an 8-bit register for monitoring in the user mode the output data from the 128K byte flash memory (internal ROM) and the 4K byte flash memory (CROM) read out in the serial/parallel mode. When the security function has been enabled, this register is for confirming in the user mode that the data read out from the internal ROM and the CROM in the serial or parallel mode is "00H". When the security function has been enabled, the data read out in the serial or parallel mode can be verified by successively executing a ROM table reference instruction followed by an instruction reading out this FLABUF register. In the user mode, if a ROM table reference instruction is executed, the data at the target address of the flash memory is transferred to FLABUF. When an instruction reading out FLABUF is executed next, the data "00H" is returned irrespective of the contents of the flash memory, thereby confirming that the security function has been enabled and that it is not possible to read out the contents in the serial or parallel modes. Execute the ROM table reference instruction and the instruction reading out FLABUF in the byte access mode.

Further, set the target ROM of ROM table reference instruction in bit 11 (TLPC) of FLAADRS). In addition, even when the security function has been enabled, it is possible to read out the contents of the flash memory using a ROM table reference instruction. The configuration of FLABUF is shown in Figure 2-9.

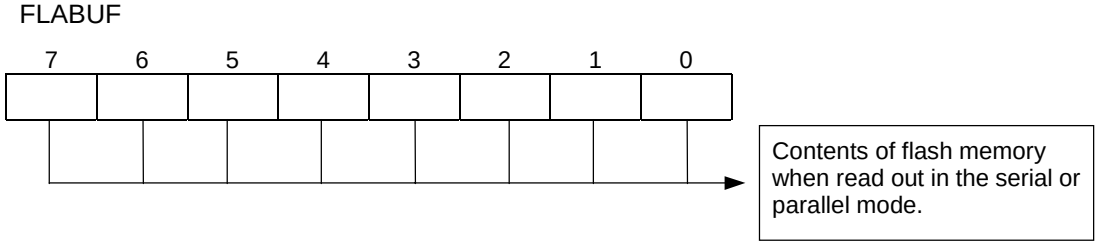


Figure 2-9 The configuration of FLABUF

(5) Hold mode

The CPU goes into the hold state (the hold mode) after one instruction is executed, when either programming, or entire chip erasing, or security bit setting is done by setting the PRG flag (bit 0), the SEPRG flag (bit 1), the CERS flag (bit 2) or the SEQS flag (bit 6) of FLACON is set (to "1").

In the hold mode, the source clock will be operating, and even the TBC, the flexible timer, the serial port, etc., will be operating. The watchdog timer (WDT) will not stop during the hold mode for programming the flash memory, so do not activate the watchdog timer (WDT). Although interrupts are accepted during the hold mode, their service will be kept pending.

Further, since the supply of clock to the CPU section will have been stopped, instruction execution is not made and will have stopped at the leading instruction immediately after the one in which the hold mode was initiated. When the programming, entire chip erasing, or security bit setting is completed, the hold state is released automatically. At this time, the interrupt servicing is executed if already there is an interrupt request and if the interrupt has been enabled, otherwise, the execution is restarted from the instruction immediately following the one in which the hold mode was initiated.

The conditions of the output pins, etc., during the hold mode are shown in Table 2-3.

Table 2-3 List of conditions in the hold mode

CPU		Stopped
Output pins	P0 to P12	No change
Operation of built-in functions	TBC	Operating
	WDT	Operating (when activated)
	FTM	Operating
	GTMC, GEVC	Operating
	S0TM to S4TM	Operating
	SCI0 to 5	Operating
	ADC	Operating
	PWM	Operating
	Extended port	Operating

(6) Examples of flow chart in the user mode
The flow chart of programming in the user mode is shown in Figure 2-10.
Further, a sample flow chart of entire chip erasure is shown in Figure 2-11, a sample flow chart of programming 128K bytes is shown in Figure 2-12, a sample flow chart of programming with auto-erase is shown in Figure 2-13, and a sample flow chart of programming is shown in Figure 2-14.

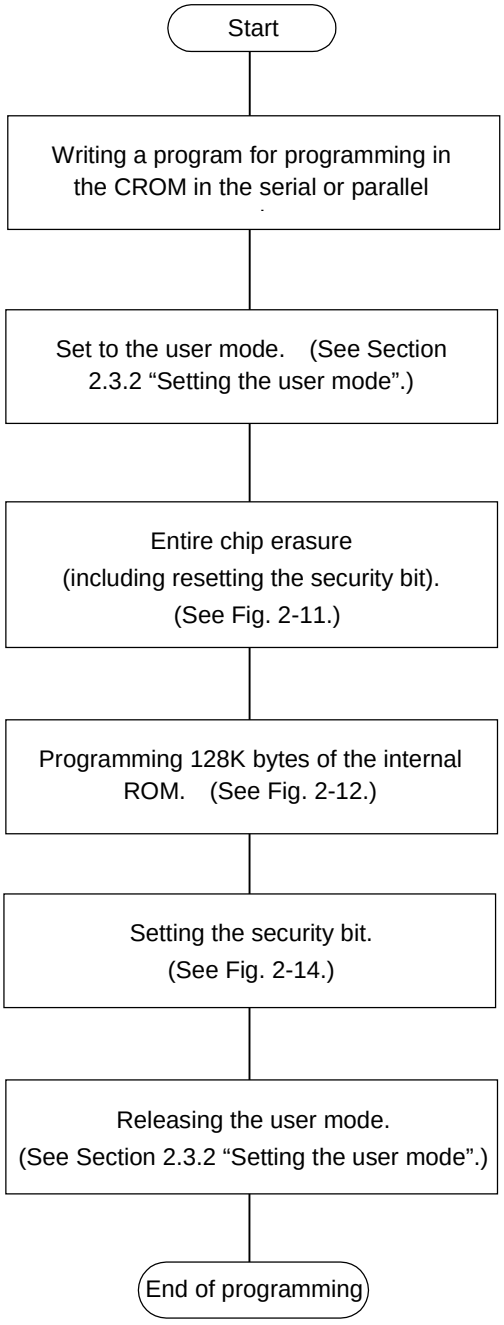
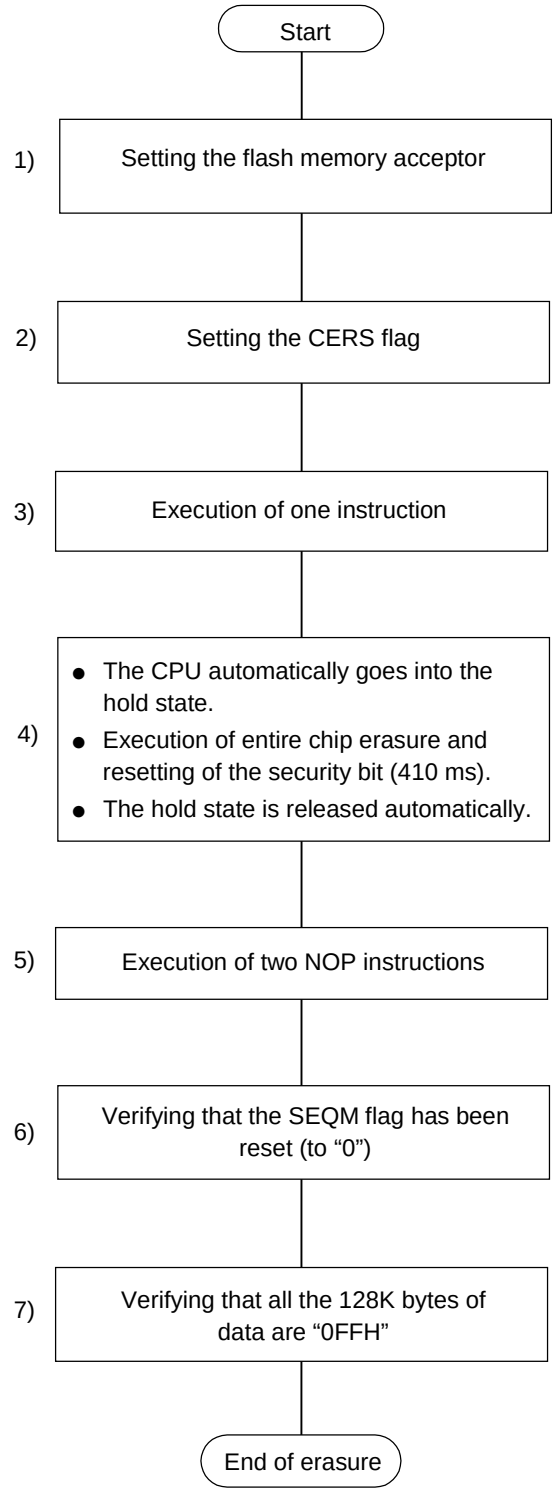


Figure 2-10 Flow chart of programming in the user mode



1) Set (to "1") the flash memory acceptor by successively writing n5H, nAH (n = 0 to F) in FLAACP.

2) Set (to "1") the entire chip erase (CERS) flag of FLACON. The CERS flag and the flash memory acceptor are reset (to "0") when a programming or erasure of the internal ROM is executed.

3) The CPU executes one instruction after the CERS flag is set. (for example, an NOP instruction)

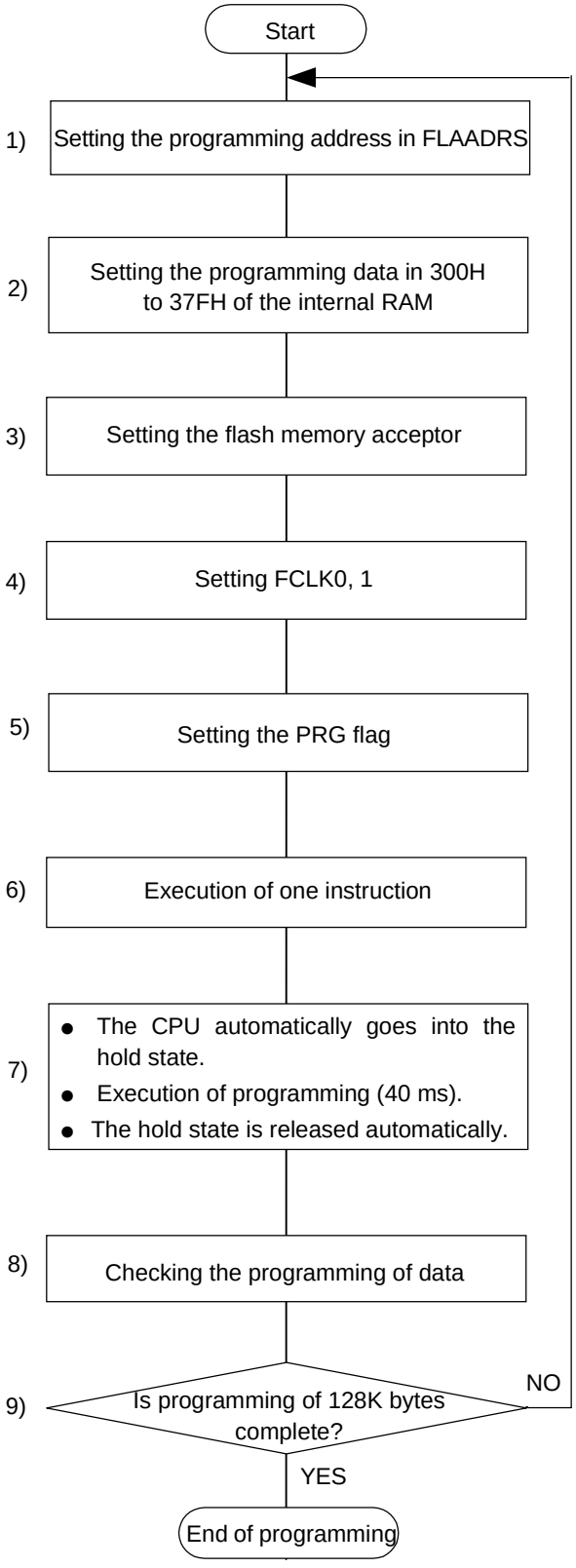
4) The CPU automatically goes to the hold state after executing one instruction.
When the CPU goes into the hold state, the entire chip erasure of the internal ROM and resetting of the security bit are carried out.
When the entire chip erasure of the internal ROM and the resetting of the security bit are completed, the hold state is released automatically, and the CPU executes the next instruction.

5) Insert two NOP instructions when reading out the SEQM flag immediately after the hold state is released.

6) Verify that the security monitoring flag (SEQM) has been reset (to "0").

7) The data "0FFH" is returned when the data of the entire 128K byte of the internal ROM is read out. The ROM table reference instruction is used for verifying the data.

Figure 2-11 Sample flow chart of entire chip erasure



1) Set the higher order 10 bits (FA16 to 7) of the programming location in units of 128 bytes in FLAADRS.

2) Set the data for programming in the addresses 300H to 37FH of the internal RAM using an UART, etc.

3) Set (to "1") the flash memory acceptor by successively writing n5H, nAH (n = 0 to F) in FLAACP.

4) Set the clock for transferring the data for programming from the internal RAM to the internal ROM. Also make the setting so that the transfer clock is 5 MHz or less.

5) Set (to "1") the write flag (PRG) of FLACON. The PRG flag and the flash memory acceptor are reset (to "0") when an erasure or a programming of the internal ROM is carried out.

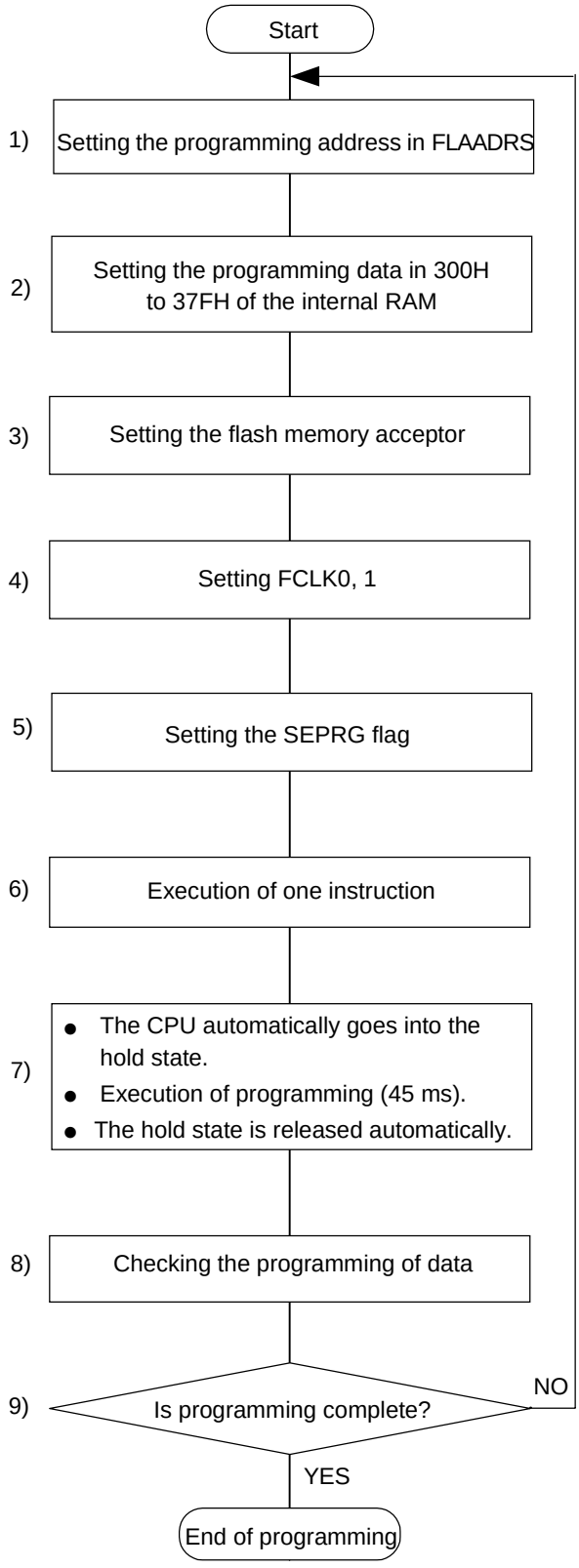
6) The CPU executes one instruction after the PRG flag is set (for example, an NOP operation).

7) The CPU automatically goes to the hold state after executing one instruction.
When the CPU goes into the hold state, the data in the addresses 300H to 37FH of the internal RAM is transferred to the internal ROM thereby carrying out programming.
When the programming of the internal ROM is completed, the hold state is released automatically, and the CPU executes the next instruction.

8) Verify that the data of the internal ROM has been programmed. The ROM table reference instruction can be used for verifying the data.

9) Since the programming is done in units of 128 bytes, it is necessary to program 1,024 times after an entire chip erasure.

Figure 2-12 Sample flow chart of programming 128K bytes



1) Set the higher order 10 bits (FA16 to 7) of the programming location in units of 128 bytes in FLAADRS.

2) Set the data for programming in the addresses 300H to 37FH of the internal RAM using an UART, etc.

3) Set (to "1") the flash memory acceptor by successively writing n5H, nAH (n = 0 to F) in FLAACP.

4) Set the clock for transferring the data for programming from the internal RAM to the internal ROM. Also make the setting so that the transfer clock is 5 MHz or less.

5) Set (to "1") the write flag (SEPRG) of FLACON. The SEPRG flag and the flash memory acceptor are reset (to "0") when an erasure or a programming of the internal ROM is carried out.

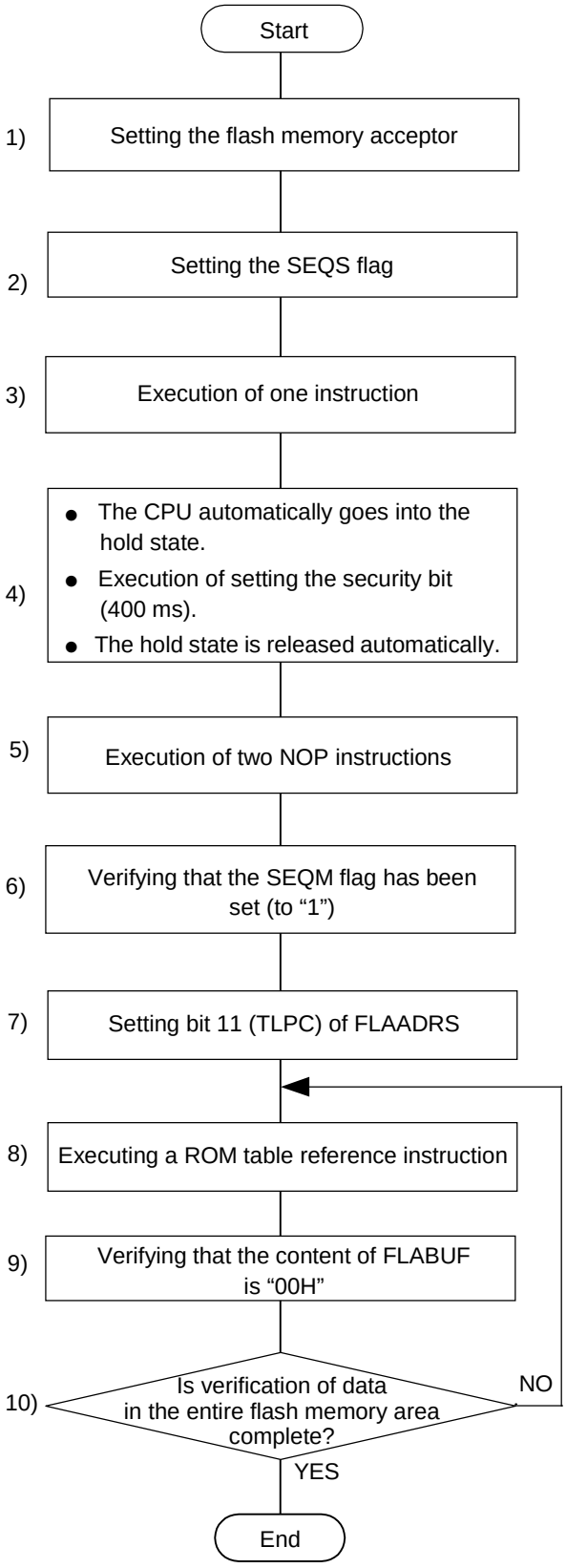
6) The CPU executes one instruction after the SEPRG flag is set (for example, an NOP operation).

7) The CPU automatically goes to the hold state after executing one instruction.
When the CPU goes into the hold state, the 128 bytes of data in the internal ROM specified by FLAADRS is erased, and the data in the addresses 300H to 37FH of the internal RAM is transferred to the internal ROM thereby carrying out programming with the auto-erase function.
When the programming of the internal ROM is completed, the hold state is released automatically, and the CPU executes the next instruction.

8) Verify that the data of the internal ROM has been programmed. The ROM table reference instruction can be used for verifying the data.

9) Since the programming is done in units of 128 bytes, if it is necessary to program more than 128 bytes, carry out the programming with the auto-erase function again for the number times necessary.

Figure 2-13 Sample flow chart of programming with auto-erase function



1) Set (to "1") the flash memory acceptor by successively writing n5H, nAH (n = 0 to F) in FLAACP.

2) Set (to "1") the security (SEQS) flag of FLACON. The SEQS flag and the flash memory acceptor are reset (to "0") when an erasure of the internal ROM or a setting of the security bit are executed.

3) The CPU executes one instruction after the SEQS flag is set. (for example, an NOP instruction)

4) The CPU automatically goes to the hold state after executing one instruction.
When the CPU goes into the hold state, the setting of the security bit is carried out.
When the setting of the security bit is completed, the hold state is released automatically, and the CPU executes the next instruction.

5) Insert two NOP instructions when reading out the SEQM flag immediately after the hold state is released.

6) Verify that the security monitoring flag (SEQM) has been set (to "1").

7) Select either the internal ROM or the CROM as the target for verifying the security status by setting bit 11 (TLPC) of FLACON appropriately.

8) Execute a ROM table reference instruction (byte access only), and transfer the data of the target address of the flash memory (the internal ROM or the CROM) to FLABUF.

9) Verify that the data of the target address is "00H" by reading out FLABUF.

10) By reading out FLABUF, it is possible to verify that the contents of the entire area of flash memory (internal ROM or CROM) are "00H" due to setting the security bit.

Figure 2-14 Sample flow chart of setting the security bit

(7) Programming precautions in the user mode
Take the following precautions when preparing the CROM program used in the user mode.

- It is not possible to write data into or erase the data in the CROM in the user mode. Carry out these operations in the parallel mode or in the serial mode.
- The CPU executes the program taking CROM as the addresses 0000H to 0FFFH (4K bytes) of segment 0 of the flash memory space.
- The CPU executes a reset based on a BRK instruction when the program address of CROM becomes 1000H to 0FFFFH of segment 0 or any address in segment 1.
- With the bit 11 (TLPC) of FLAADRS in the "1" state, always the value "0FFH" is returned if an address in the range 1000H to 0FFFFH of segment 0 or any address in segment 1 is accessed using a ROM table reference instruction.
- After setting the user mode, carry out the first programming to the internal ROM only after carrying out an entire chip erasure.
- Limit successive entire chip erasure or programming based on PRG to the same address to two operations.
- Although the data becomes "0FFH" when the flash memory is erased, even writing a data of "0FFH" thereafter should be counted as one programming operation.
- The instruction after setting (to "1") the PRG, SEPRG, CERS, or SEQS flags of FLACON should be an instruction other than the instructions setting the standby mode (STOP, HALT).
- When two or more of the PRG, SEPRG, CERS, or SEQS flags of FLACON are set (to "1") simultaneously, the CPU will not go into the hold mode even after one instruction is executed, and hence programming or erasing will not be done. In such a case, the next instruction is executed after executing one instruction.
- When the security function is enabled by setting (to "1") the security bit, the PRG and SEPRG flags of FLACON will not be set (to "1"). In this case, the CPU does not go into the hold state and the next instruction is executed after executing one instruction. When carrying out a programming by setting (to "1") the PRG or the SEPRG flag of FLACON, carry out an entire chip erasure by setting (to "1") the CERS flag, and reset (to "0") the security bit thereby disabling the security function.
- Before executing a ROM table reference instruction, always set bit 11 (TLPC) of FLAADRS to "0" or "1" thereby specifying the target ROM.
- When reading out the FLABUF register, after executing a ROM table reference instruction, execute an instruction reading out FLABUF. Also, execute the ROM table reference instruction and the instruction reading out FLABUF as byte operation instructions.
  The read out data will be indefinite if only the instruction reading out FLABUF is executed.
- The CPU refers to the vector table of CROM, when a reset due to a $\overline{\text{RES}}$ pin input with the $\overline{\text{EA}}$ pin and the TEST pin at the H level, a reset due to the execution of a BRK instruction, a reset due to an overflow of the watch dog timer (WDT), a reset due to an operation code trap (OPTRP), or any of the various types of interrupt requests are generated when the user mode has been initiated (that is, when the CROM program is being executed).
- When the user mode is set by mistake, the CPU executes the CROM program, and there is the possibility of the contents of the 128K byte flash memory (the internal ROM) get programmed accidentally.
  Therefore, set a program in the CROM that always asks for permission to execute from a device external to the MSM66Q591 before executing each operation of carrying out entire chip erasure or programming of the 128K byte flash memory, or setting the security bit.
- Insert protection circuits when connecting the user mode setting pins to external devices.
- The watchdog timer (WDT) will not stop during the hold mode for programming the flash memory, erasing the entire chip, and setting the security bit, so do not activate the watchdog (WDT) timer.

2.4 Serial Mode

2.4.1 Overview of serial mode

The serial mode is one in which the programming is possible with the chip installed in the system similar to the user mode.

The programming is done using either a flash memory writer dedicated to the serial mode (the Flash Memory Programmer (FW66500S) manufactured by OKI Electric) or a general purpose flash memory programmer (the general purpose flash memory microcomputer programmer (AF200) manufactured by Yokogawa Digital Computers). There is no need to prepare a write program (CROM) that is different from the one for the user mode.

In addition, in the serial mode, the programming is done when the microcomputer is either in the reset state or in the stop mode. During serial mode programming in the stop mode, the contents of the internal RAM, timer, or other peripherals will not be changed.

Further, for details of the flash memory writer, see the users manual of the Flash Memory Programmer (FW66500S) manufactured by OKI Electric or of the general purpose flash memory microcomputer programmer (AF200) manufactured by Yokogawa Digital Computers.

2.4.2 Settings of serial mode

The serial mode is set automatically when programming or reading out is executed after connecting the flash memory programmer to the specific pins, and gets released when the programming or reading out is completed.

(1) Pins used in the serial mode

The pins used in the serial mode are shown in Table 2-4.

The serial mode can be set only while the microcomputer is either in the reset state or in the stop mode. Take care that a high voltage is applied to the TEST pin from the flash memory programmer to set to the serial mode. The V_{DD} pin is connected for V_{DD} monitoring of user application system.

Insert protection circuits into P11_0 and P11_2 pins.

Table 2-4 Pins used in the serial mode

Pin No.	Primary function	Flash memory function name
95	P11_2	FLACLK (Serial clock input)
93	P11_0	FLADAT (Serial data I/O)
92	TEST	FLAMOD (High voltage input for serial mode setting)
3	$\overline{EA}$	Input for serial mode setting
4, 41, 61, 80, 105, 127	V_{DD}	V_{DD} monitoring of user application system
32, 44, 62, 81, 106, 128	GND	Ground

Caution: During the serial mode, a voltage higher than the power supply voltage is applied to the TEST pin.

(2) Serial mode connection circuit

In the serial mode, it is necessary to connect the flash memory programmer FW66500S or AF200 to the pins P11_0, P11_2, TEST, V_{DD}, and GND of the MSM66Q591 in the user's application system. In addition, since the $\overline{EA}$ pin is an input pin for setting the serial mode, it should be connected to the V_{DD} pin. Also, provide switches in the user's application system for isolating the signal lines of the MSM66Q591 from the user's application system during writing or reading in the serial mode.

An example of the serial mode connection circuit, Example 1, is shown in Figure 2-15.

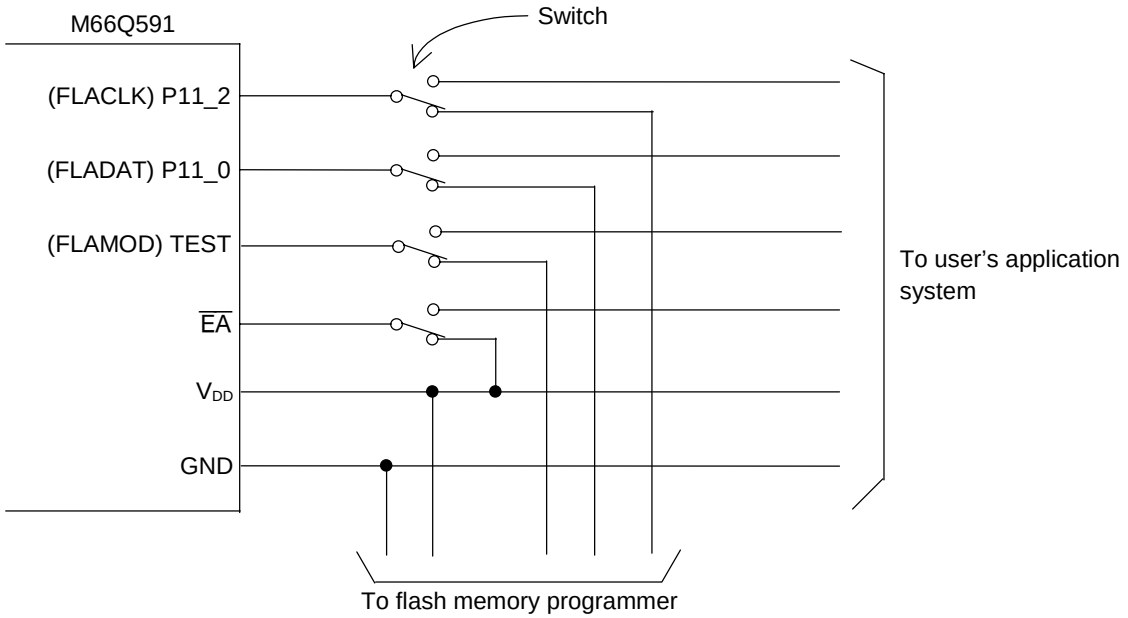


Figure 2-15 Example 1 of serial mode connection circuit

If it is not possible to provide switches in the user's application system, do not use the pins P11_0 and P11_2 in the user's application system but connect them only to the flash memory programmer. Further, pull up each of the pins P11_0 and P11_2 to V_{DD} via resistors of about 100 kΩ, and pull down the TEST pin to GND via a resistor of about 10 kΩ. Connect the $\overline{EA}$ pin to V_{DD}. Also, note that these resistance values are merely for guidance and sufficient evaluation should be made before deciding on the actual resistance values.
An example of the serial mode connection circuit, Example 2, is shown in Figure 2-16.
Use a resistor of 1 kΩ or less if connecting an external resistor in series with the TEST pin (Point(A)).

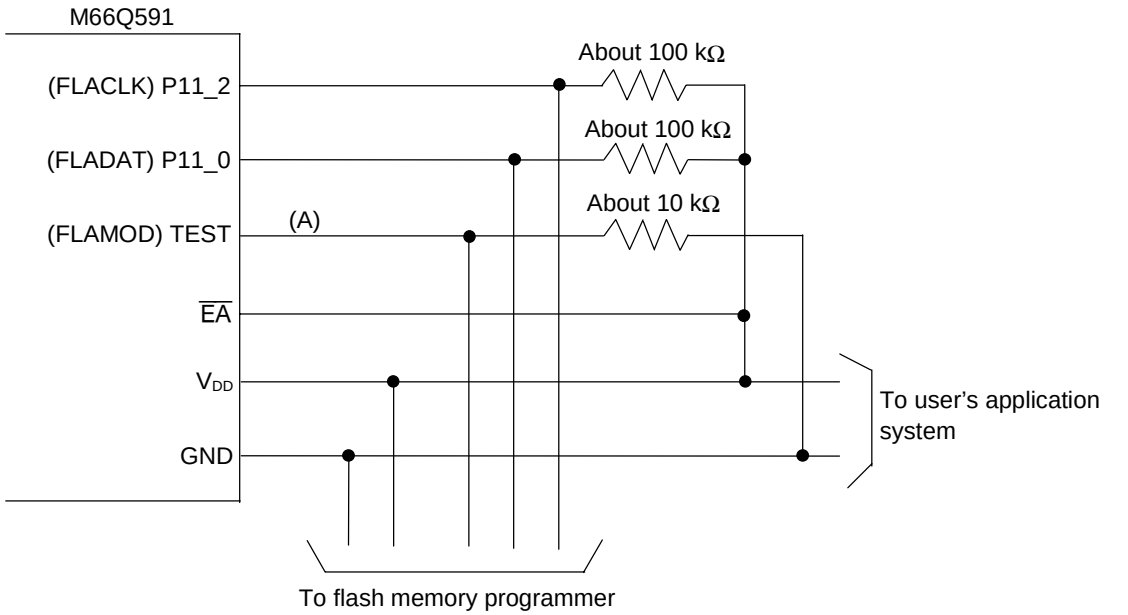


Figure 2-16 Example 2 of serial mode connection circuit

2.4.3 Programming in the serial mode

In the serial mode, programming is done using a flash memory programmer, the FW66500S or the AF200. The procedure of programming using a flash memory programmer is given below.

1) Connect the flash memory programmer FW66500S or AF200 to the pins P11_0, P11_2, TEST, V_{DD}, and GND of the MSM66Q591. Also, connect the $\overline{EA}$ pin to the V_{DD} pin.
2) Put the microcomputer in the reset state or in the stop mode.
   - The flash memory programmer generates a protocol error when the CPU is not in either the reset state or the stop mode.
3) Carry out programming or reading out from the flash memory programmer.
   - The serial mode is set automatically.
   - During the serial mode, the servicing of the request from an NMI interrupt, an external interrupt 0, or an external interrupt 1 will be kept pending.
   - Never release the reset state during the serial mode, because doing so will cause the serial mode to be released immediately. If the reset state is released, carry out writing of the program for the entire area of the flash memory.
   - Always continue to keep the $\overline{EA}$ pin at the H level during the serial mode, because the serial mode gets released immediately if an L level is detected on the $\overline{EA}$ pin during the serial mode.
4) Confirm the normal end of operation of the flash memory programmer.
   - The serial mode is released automatically.
5) Carry out releasing of the reset state or the stop mode.
   - If a request from an NMI interrupt, an external interrupt 0, or an external interrupt 1 had been generated during programming or reading out, the stop mode will be released at the end of programming or reading out.

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Before using the product, please make sure that the information being referred to is up-to-date.

2. The outline of action and examples for application circuits described herein have been chosen as an explanation for the standard action and performance of the product. When planning to use the product, please ensure that the external conditions are reflected in the actual circuit, assembly, and program designs.

3. When designing your product, please use our product below the specified maximum ratings and within the specified operating ranges including, but not limited to, operating voltage, power dissipation, and operating temperature.

4. Oki assumes no responsibility or liability whatsoever for any failure or unusual or unexpected operation resulting from misuse, neglect, improper installation, repair, alteration or accident, improper handling, or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified operating range.

5. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of the product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.

6. The products listed in this document are intended for use in general electronics equipment for commercial applications (e.g., office automation, communication equipment, measurement equipment, consumer electronics, etc.). These products are not authorized for use in any system or application that requires special or enhanced quality and reliability characteristics nor in any system or application where the failure of such system or application may result in the loss or damage of property, or death or injury to humans.
   Such applications include, but are not limited to, traffic and automotive equipment, safety devices, aerospace equipment, nuclear power control, medical equipment, and life-support systems.

7. Certain products in this document may need government approval before they can be exported to particular countries. The purchaser assumes the responsibility of determining the legality of export of these products and will take appropriate and necessary steps at their own expense for these.

8. No part of the contents contained herein may be reprinted or reproduced without our prior permission.