Chapter 8 Alarm/Wake Feature

8.1 General

The MAPLE is furnished with a 7508 4-bit CPU which controls the software timer (clock) and generates interrupts to the Z80 CPU at specified intervals. The software timer is supported by the alarm/wake OS feature. The alarm/wake feature is divided into the following three functions:

1) Alarm function
2) Wake1 function
3) Wake2 function

These functions are identified by software using a flag; only one type of interrupt is generated by the 7508 CPU for these functions. The 7508 checks for an alarm/wake time every 10 seconds even if the MAPLE is in the power off state, that is, the alarm/wake feature remains available when MAPLE power is off. However, the alarm/wake processing differs depending on whether MAPLE power is off or on. The next section explains how alarm/wake processing proceeds in both power off and on states (refer to "OS Specifications" for details).

8.2 Alarm Function

1) What to set

(1) Alarm time (month/day/hour/minute/second (10-second units))

(2) Alarm message (up to 40 alphanumeric, kana, and graphics characters)

2) How to set

(1) Use the System Display (second cannot be specified).

(2) Use the BIOS TIMDAT function (see Chapter 4, "BIOS Calls").

(3) Load the work areas time data and issue a time setting command directly to the 7508 CPU (see Section 8.7 and Chapter 11, "7508 Explanations").

3) Alarm function in power-on state

The alarm function sounds an alarm and displays the time and message using the VRAM system screen. This guarantees that no user data on the screen be destroyed. When the display is ended, the user data displayed immediately before the alarm message is restored.

The time display can be terminated when:

(1) The ESC key is pressed.

(2) 50 seconds has elapsed.

(3) The POWER switch is turned off.

(4) A power failure occurs.

4) Alarm function in power-off state

The alarm function performs the same operations as in the power-on state after the MAPLE is powered on. After the display is terminated, the original screen before power is turned on is restored. If power is switched off and back on again while the alarm function is displaying the alarm time and message, then the normal power-on sequence occurs.

8.3 Wakel Function

1) What to set

(1) Wake time (month/day/hour/minute/second (10-second units))

(2) The name of program to be executed when a wake condition occurs.

2) How to set

Same as the alarm function in 8.2.

3) Wakel function in power-on state

The wake function treats the wake string as an alarm string and performs the same operations as the alarm function.

4) Wakel function in power-off state

When power is turned on, the wake function loads the wake string into the key buffer for execution as power-on commands.

- When the MAPLE is in the restart mode power-off state

  The wakel function executes WBOOT and displays the Menu, then enables the wake string for execution under CCP control.

  (In Overseas version 2.3 (B), the function enables the wake string for execution under CCP without displaying the menu).
- When the MAPLE is in the continue mode power-off state

  The wakel function ignores the wake string and returns the MAPLE into the state before it is powered off and continues processing.

The wakel function, when used with the BIOS POWEROFF function, may find many applications in periodic data collection and other automatic (unattended) operations without operator's intervention.

8.4 Wake2 Function

1) What to set

(1) Wake time (month/day/hour/minute/second (10-second units))

(2) The address of the routine to be executed when a wake condition occurs.

2) How to set

(1) Use the BIOS TIMDAT function (see Chapter 4, "BIOS Calls").

(2) Load the work areas time data and issue a time setting command directly to the 7508 CPU (see Section 8.7 and Chapter 11, "7508 Explanations").

3) Wake2 function in power-on state

The wake2 function calls the specified address. See programming note 5) below for the routine to be specified at this address.

4) Wake2 function in power-off state

- When the MAPLE is in the restart mode power-off state

  When power is turned on, the wake2 function calls the specified address, then returns the MAPLE into the state (restart mode) before power is turned off.

- When the MAPLE is in the continue mode power-off state

  The wake2 function returns the MAPLE into the state before it is powered off, then causes a jump to the specified address. If the destination of the jump is a RET instruction, control is returned to the point in the program at which the MAPLE was powered off in the continue mode.

5) Wake2 function programming notes

(1) Neither BDOS nor BIOS system call can be used in the routine to be called by the wake2 function.

(2) The routine to be executed by the wake2 function must end with a RET instruction.

(3) When the wake2 function is invoked in the power-off state, only power to the main board is turned on and no power is supplied to the I/O devices (e.g., RS-232C, serial port, and ROM capsules). Furthermore, if this condition occurs in the continue mode, the routine to be called by the wake2 function must turn on the power to these devices before executing the RET instruction. See the next page for the procedure for turning on the power to the I/O devices.

(4) The event which called the wake2 function can be identified by examining the following work areas:

- ZSTARTFG: Overseas version = 0F389H

  Japanese-language version = 0F0C9H

  Identifies the source of the invocation of the routine.

  01H: POWER switch on.

  02H: Alarm

  03H: Wake1

  04H: Wake2

- CNTNFG: Overseas version = ØF33ØH

  Japanese-language version = ØFØ5ØH

  Identifies the power-off state mode.

  = 00H: Continue mode.

  ≠ 00H: Restart mode.

ZSTARTFG	CNTNFG	State from which control is passed to the routine via wake2 function
≠Ø4H	NC	Power-on state.
=Ø4H	ØØH	Continue mode power-off state
=Ø4H	≠ØØH	Restart mode power-off state

The routine called by the wake2 function must examine the above work areas to identify the power-off state and immediately set the work areas as follows:

ZSTARTFG = ØØH

CNTNFG = ØFFH

When called in the continue mode power-off state, the routine must execute the following code before executing the RET instruction:

```
        LD    HL,(ATSOTIME)      ┐
        LD    DE,(TIMERØ)        │  Sets the new auto
                                 │  shut-off time.
        ADD   HL,DE              │
        LD    (TIMEEND),HL       ┘

        LD    A,(SPOPN)          ┐
        OR    A                  │  Checks to determine
        JR    NZ,AAAA            │  whether serial or
                                 │  RS-232C interface has
        LD    A,(RSCLSF)         │  been used and, if so,
        OR    A                  │  turns on its driver.
        JR    NZ,BBBB            ┘

 AAAA:                           ┬─ Turns on the driver.
        LD    A,(CTLR2)          ↓
        SET   4,A
        OUT   (2),A
        CALL  ST1ML ←───────────── 1m-second software timer.
        SET   3,A
        OUT   (2),A
        CALL  ST1ØØML ←─────────── 1ØØm-second software
                                   timer.
        RES   4,A
        SET   5,A
        LD    (CTLR2),A
        OUT   (2),A
```

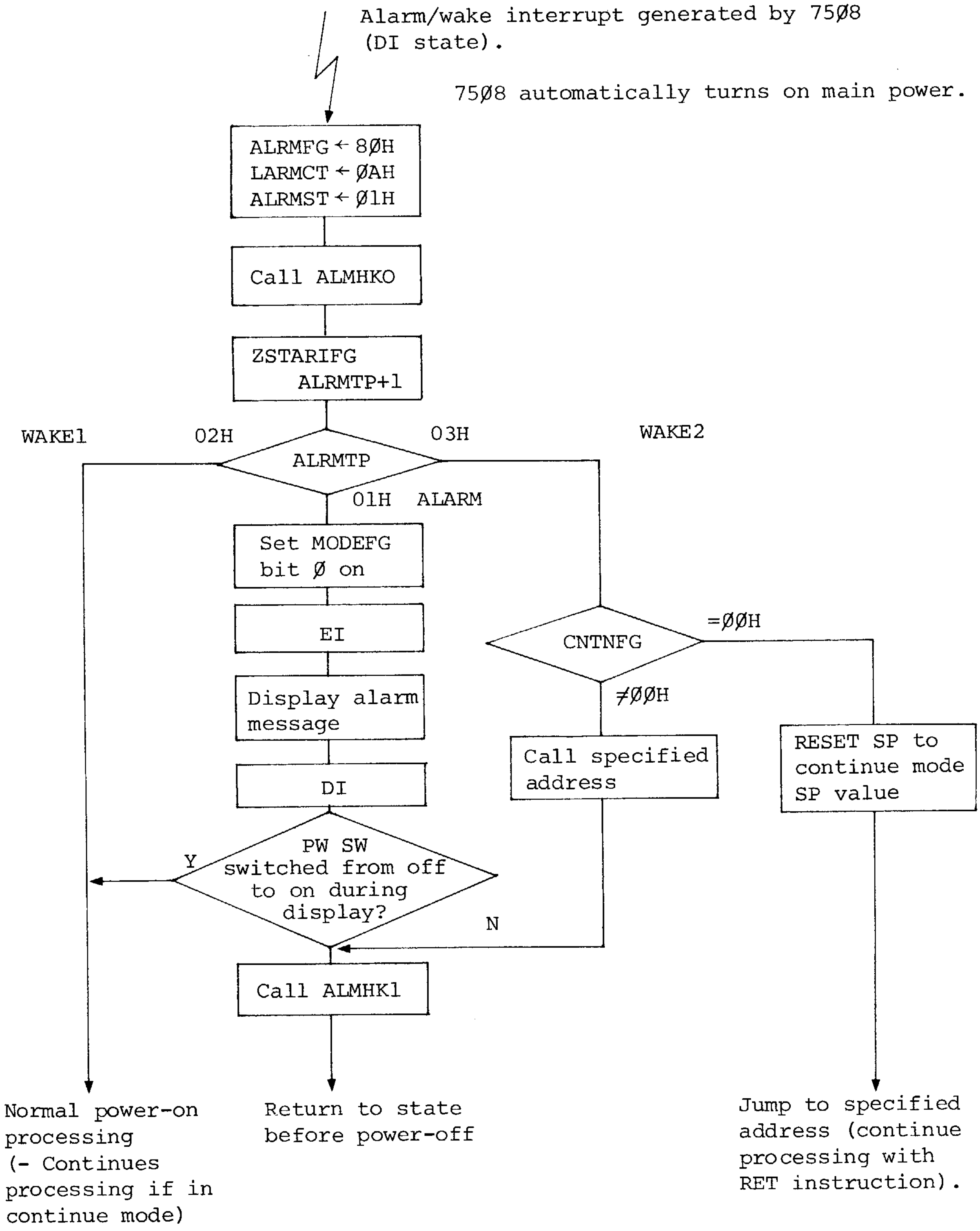
```
BBBB:
        LD      A,(RSCLSF)
        OR      A
        JR      NZ,CCCC

        LD      A,ØBEH
        OUT     (ØDH),A                 Initializes 8251 if
        LD      A,Ø4ØH                  the RS-232C interface
        OUT     (ØDH),A                 has been used.
        CALL    ST1ØØML
        LD      A,(SVRSMOD)
        OUT     (ØDH),A
        LD      A,(SVRSCMD)
        OUT     (ØDH),A


CCCC:
        XOR     A                       Turns off ROM capsule
        LD      (PROMPWR),A             power.



        LD      HL,MTIMEBUF
        LD      DE,YPOFDS
        LD      BC,4
        LDIR

        LD      A,(IER)
        OUT     (Ø4H),A

        LD      A,(CNTNILVL)
        LD      (INTLEVEL),A

        POP     HL
        POP     DE                      Restores registers.
        POP     BC
        POP     AF

        EX      AF,AF'
        EXX
        POP     IY
        POP     IX

        RET
```

Work area address chart

Work area name	Overseas version	Japanese-language version
ATSOTIME	FØ27H	ED27H
TIMERØ	FØ71H	ED72H
TIMEEND	F6DCH	F46BH
SPOPN	F35AH	FØ82H
RSCLSF	F2C8H	EFF8H
CTLR2	FØB2H	ED92H
SVRSMOD	F6DØH	F45DH
SVRSCMD	F6D1H	F45CH
PROMPWR	F1CAH	EEE3H
MTIMEBUF	F4BDH	F232H
YPOFDS	FØD9H	EDB9H
IER	FØB3H	ED93H
CNTNILVL	F385H	FØC5H
INTLEVEL	FØBAH	ED9AH

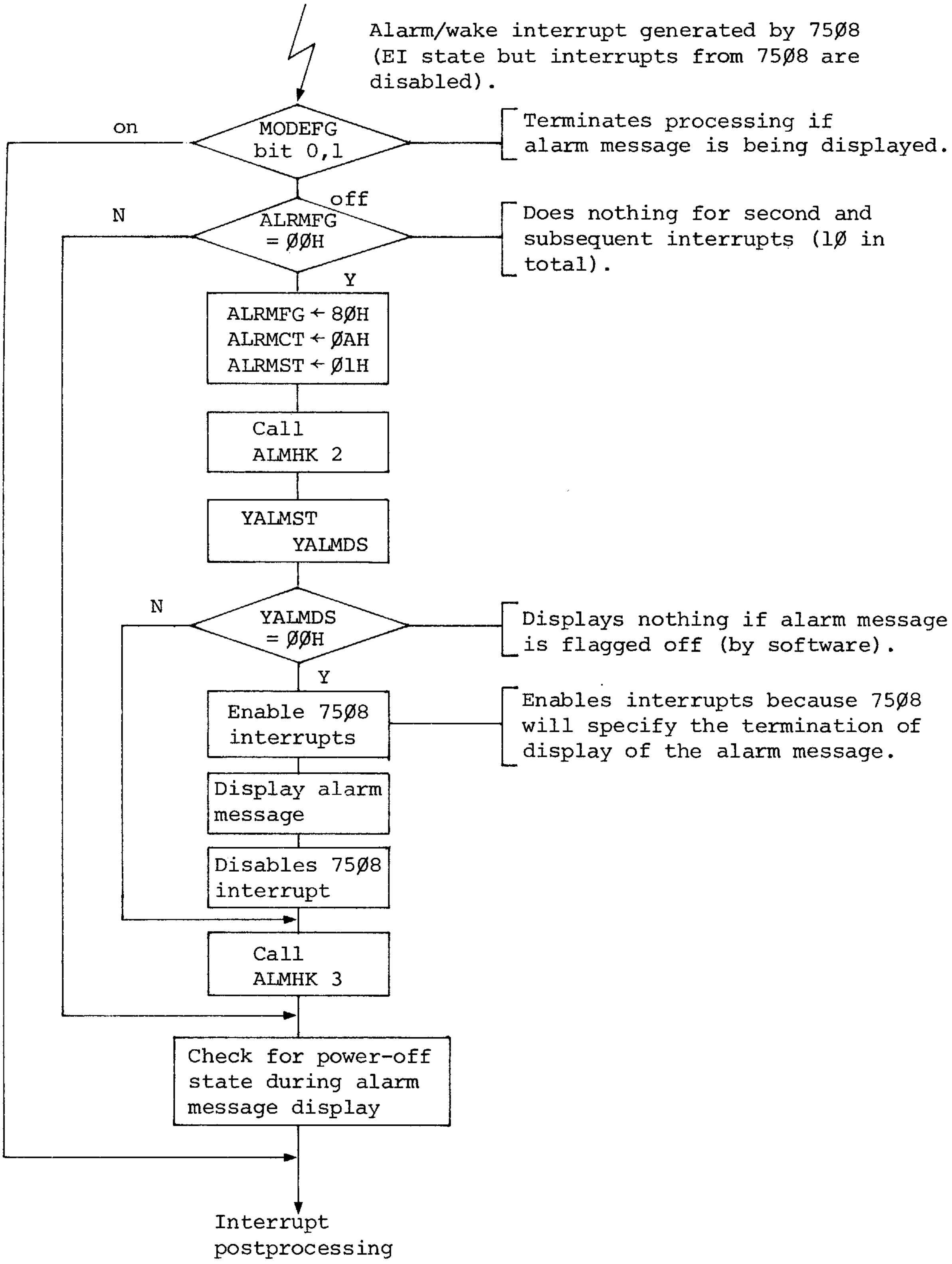
8.5 Alarm/Wake Function Processing Flow

1) Alarm/Wake processing in the power-off state



Alarm/wake interrupt generated by 75Ø8 (DI state).
75Ø8 automatically turns on main power.
ALRMFG ← 8ØH
LARMCT ← ØAH
ALRMST ← Ø1H
Call ALMHKO
ZSTARIFG
ALRMTP+1
WAKE1
02H
ALRMTP
03H
WAKE2
01H ALARM
Set MODEFG bit Ø on
EI
Display alarm message
DI
PW SW switched from off to on during display?
Y
N
Call ALMHK1
CNTNFG
=ØØH
≠ØØH
Call specified address
RESET SP to continue mode SP value
Normal power-on processing (- Continues processing if in continue mode)
Return to state before power-off
Jump to specified address (continue processing with RET instruction).

2) Alarm/Wake processing in the power-on state



Alarm/wake interrupt generated by 75Ø8 (EI state but interrupts from 75Ø8 are disabled).

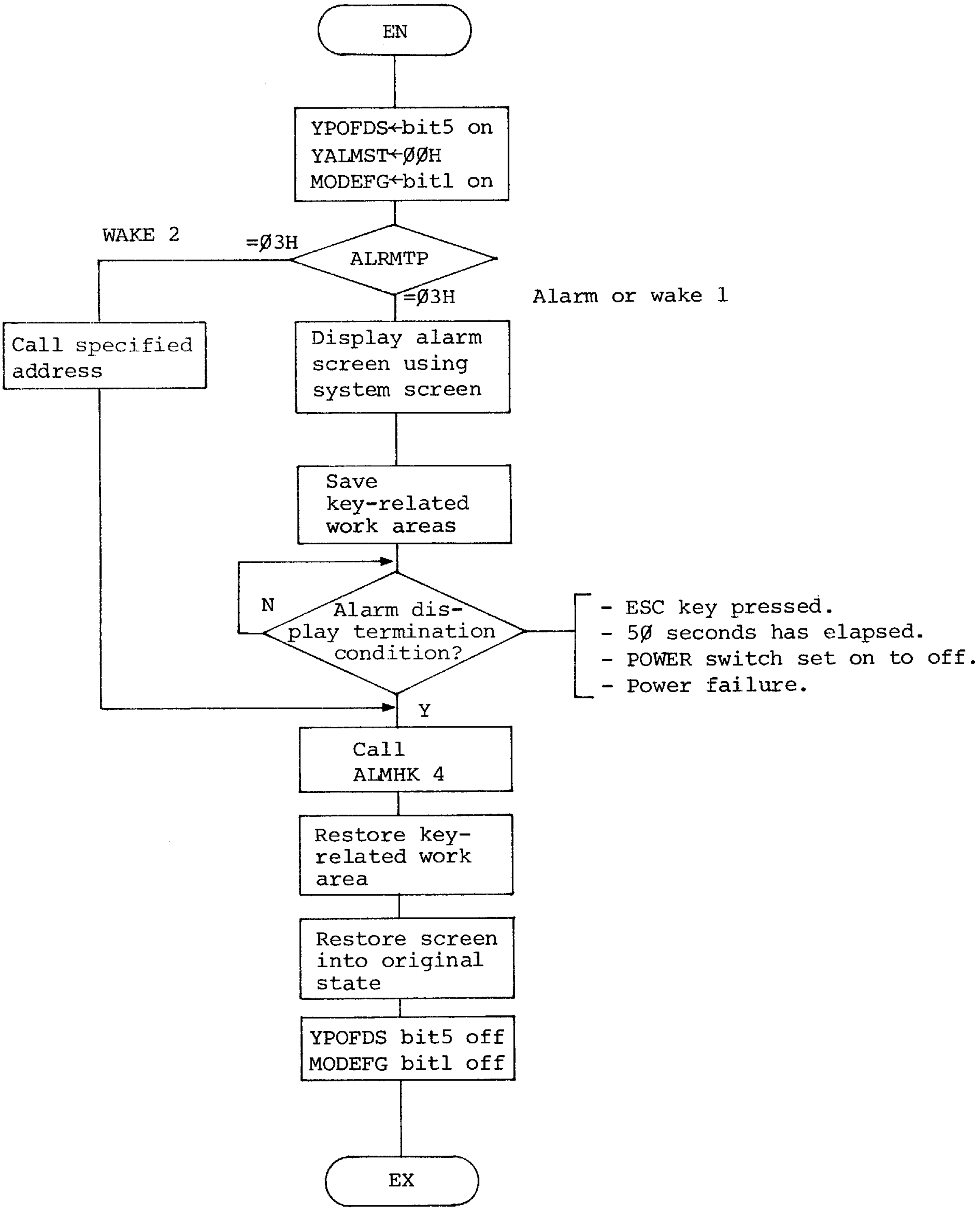
Terminates processing if alarm message is being displayed.

Does nothing for second and subsequent interrupts (1Ø in total).

Displays nothing if alarm message is flagged off (by software).

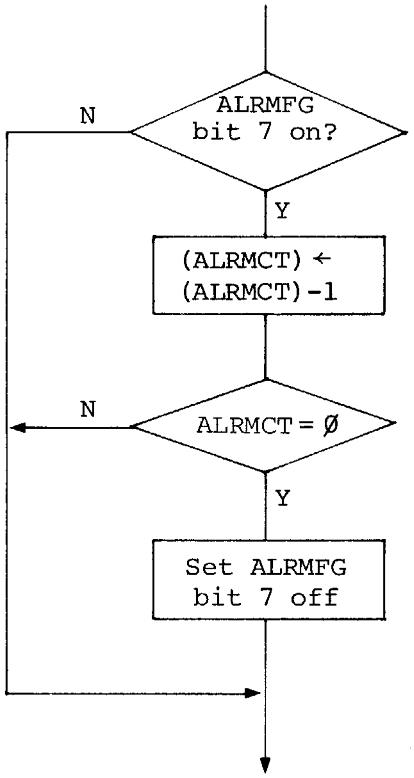
Enables interrupts because 75Ø8 will specify the termination of display of the alarm message.

3) Alarm display processing


EN
YPOFDS←bit5 on
YALMST←ØØH
MODEFG←bit1 on
WAKE 2
=Ø3H
ALRMTP
=Ø3H
Alarm or wake 1
Call specified address
Display alarm screen using system screen
Save key-related work areas
N
Alarm display termination condition?
- ESC key pressed.
- 5Ø seconds has elapsed.
- POWER switch set on to off.
- Power failure.
Y
Call ALMHK 4
Restore key-related work area
Restore screen into original state
YPOFDS bit5 off
MODEFG bit1 off
EX

4) Alarm processing during 1-second interrupt processing

Since alarm/wake interrupts occur every one second, a total of 10 times during the 10-second period, the interrupt handler ignores the second and subsequent interrupts. Accordingly, the OS examines the ALRMFG flag for 10 seconds (for 10 interrupts) since the first interrupt occurred using the 1-second interrupt processing routine and indicates the results to the alarm/wake processing routine.



The first alarm/wake interrupt sets ALRMFG and ALRMCT flags as follows:

ALRMFG = 8ØH

ALRMCT = ØAH

After 1Ø seconds, ALRMFG is set to ØØH. The interrupt handler ignores any alarm/wake interrupts while ALRMFG is nonzero.

Summary of work areas used by the alarm/wake functions

Work area name (Address)	Size (in bytes)	Description
ALRMTP (Overseas version = ØFØ6CH, Japanese-language version = ØED6DH)	1	Identifies the type of alarm/wake functions. = ØØH: Undefined = Ø1H: Alarm = Ø2H: Wake1 = Ø3H: Wake2
ALRMAD (Overseas version = ØFØ6DH, Japanese-language version = ØED6EH)	2	Contains the starting address of the alarm message or wake string.
ALRMST (Overseas version = ØFØ6FH, Japanese-language version = ØED7ØH)	1	Indicates whether an interrupt occurred or not for the currently set alarm/wake time. = ØØH: Not occurred. = ØH: Occurred.
ALRMFG (Overseas version = ØFØ7ØH, Japanese-language version = ØED71H)	1	Indicates the time count from the timer when an alarm/wake interrupt occurred (up to 1Ø seconds). Bit 7: Set by the first interrupt and cleared after 1Ø seconds.
MODEFG (Overseas version = ØFØB8H, Japanese-language version = ØED98H)	1	Indicates the current system status. Bit Ø = 1: Alarm/wake processing invoked from power-off state is in progress. Bit 1 = 1: Alarm/wake processing invoked from power-on state is in progress.

Work area name (Address)	Size (in bytes)	Description
YALMDS (Overseas version = ØFØDBH, Japanese-language version = ØEDBBH)	1	Indicates the alarm/wake disable state. Bit 7 = 1: Disabled because BIOS is in execution. Bit 6 = 1: Disabled because password is being entered. Bit 5 = 1: Disabled because alarm/wake message is being displayed. Bit 4 = 1: Disabled because system message is being displayed. Bit 3 = 1: Disabled by BASIC. Bit 2 = 1: Disabled by scheduler. Bit 1 = 1: Disabled by MTOS. Bit Ø = 1: Reserved (for applications).
YALMST (Overseas version = ØFØDCH, Japanese-language version = ØEDBCH)	1	Indicates that an alarm/wake interrupt occurred when the alarm/wake functions are disabled. The meanings of the bits are identical to those of YALMDS.
ALRMCT (Overseas version = ØF4E6H, Japanese-language version = ØF25BH)	1	Contains the number of alarm/wake interrupts. The 75Ø8 generates an interrupt every one second for 1Ø seconds (1Ø in total) for one alarm/wake time.

8.6 How to Augment the Alarm/Wake Functions Using Hooks

As shown in Section 8.5, the alarm/wake functions has five hooks. The user can extend the alarm/wake functions by making patches in these hooks. This section shows how to make patches for these hooks.

Hook addresses

	Address	Label	Contents
Overseas Ver.	ØEF8CH	ALMHKØ:	JP RETURN
Japanese Ver.	ØEBD8H		
Overseas Ver.	ØEF8FH	ALMHK1:	JP RETURN
Japanese Ver.	ØEBDBH		
Overseas Ver.	ØEF92H	ALMHK2:	JP RETURN
Japanese Ver.	ØEBDEH		
Overseas Ver.	ØEF95H	ALMHK3:	JP RETURN
Japanese Ver.	ØEBE1H		
Overseas Ver.	ØEF98H	ALMHK4:	JP RETURN
Japanese Ver.	ØEBE4H		
Overseas Ver.	ØEEB7H	RETURN:	RET
Japanese Ver.	ØEBØBH		

The above entries can be hooked to any user-supplied routines by changing the address portion of the JP RETURN instruction.

Programming notes that the user must take when changing hook addresses follow.

(1) Take care with bank control.

The hook entries are always called when the system is in the system bank state (addresses ØØØØH through 7FFFH are allocated for ROM and 8ØØØH through ØFFFFH for RAM). No problem will occur as long as the jump addresses in the hook table point to memory addresses 8ØØØH and higher; however, the active bank need be switched to the user bank whenever hook entries are entered if they point to addresses between ØØØØH and 7FFFH. Normally no user-supplied routine can be placed in addresses between ØØØØH and 7FFFH.

(2) Reserve a user stack area.

Since control is transferred to the hook with the stack pointer pointing to the stack for interrupt routines, if the routine pointed to by the hook is to use a stack area (e.g., when using CALL and/or PUSH instructions), it must reserve its own stack area and restores the stack pointer to the original value when it terminates execution.

(3) Save the contents of registers and work areas.

Control is passed to the hook without saving the contents of registers and work areas. Accordingly, if a user routine specified in the hook is to alter registers or system work areas, it must save the contents of the registers and work areas to be changed on entry and restores them on exit (of course, it can safely alter the contents of work areas which expect alteration by the user).

(4) Do not change the interrupt status.

Since ALMHKØ-ALMHK4 are invoked when the CPU is in one of the interrupt states listed below, no user-supplied routine specified in the hook can change the interrupt state. If a user-supplied routine need to change the interrupt state, it must restore the MS into the original interrupt state before terminating processing.

ALMHK0:	DI state
ALMHK1:	DI state
ALMHK2:	EI state (7508 interrupts are disabled.)
ALMHK3:	EI state (7508 interrupts are disabled.)
ALMHK4:	EI state

(5) Disable interrupts when changing an address in the hook.

The system is highly likely to crash if an interrupt using a hook entry occurs while the address in that entry is being changed. Since alteration of jump addresses in the hook is normally done by the user program in the TPA, the user program can and should inhibit such interrupts with a DI instruction to avoid possible system crash. The program, however, must executes an EI instruction after terminating its execution.

(6) Do not call any system routine from the hook.
The hook does not know from what system state it is called because it is invoked by interrupts. It may be called while a system routine (BDOS, BIOS, or internal OS routine) is being executed. A system crash will occur if a routine in the hook calls a system routine in such a situation.

(7) Do not perform an I/O operation.
For the same reason given above, the routines in the hook must not perform any I/O operations such as display on the screen, communication through the RS-232C interface, etc.

(8) Since the jump table in the hook is initialized by system initialize or reset processing (placed into the state described on page 8-19), when the hook routines are to be made resident in memory, write to that effect in the manual. After system initialize or reset processing is performed, run a program for setting up the hook jump table. (Reset processing initializes only the hook jump table and keeps the user BIOS and RAM disk contents

intact.)

(9) Generally, the routines to be executed in the hook should be placed in the user BIOS area. This makes them resident in memory and solves the problem discussed in (1).

8.7 Making Alarm/Wake Settings Directly for 75Ø8

As explained in Sections 8.2, 8.3, and 8.4, alarm/wake settings can easily be made by means of System Display or BIOS calls. When alarm/wake settings are to be made in interrupt processing routines as scheduled by a scheduler, however, there is no way but to define alarm/wake information directly to the 7508 CPU for the reason given in paragraph (6) in 8.6.

The 75Ø8 subsystem is provided with the following four functions (commands) associated with the alarm/wake feature:

- ALARM (WAKE) SET
- ALARM (WAKE) READ
- ALARM (WAKE) ON
- ALARM (WAKE) OFF

See Chapter 11, "75Ø8 CPU" for details on the above functions and the interface to the 75Ø8.

This section describes the procedure for defining the alarm, wakel, and wake2 information directly to the 7508 CPU.

1) Alarm setting procedure

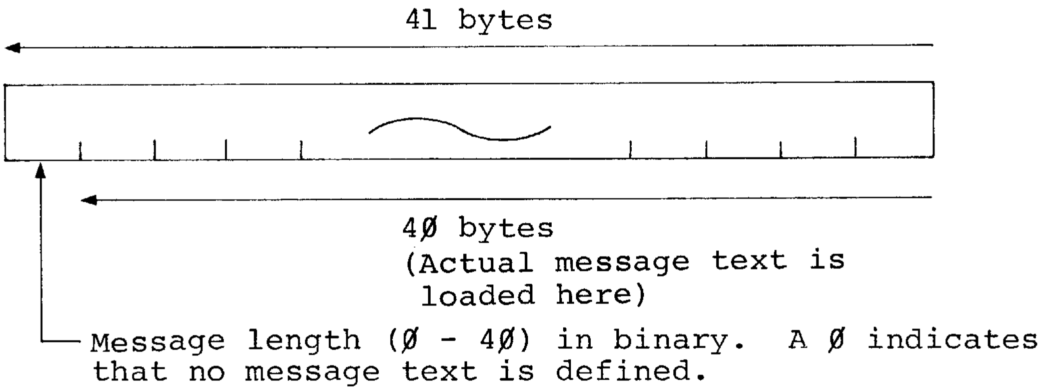
(1) Disable interrupts from the 7508.

```
     LD   A, (IER)
     RES  0, A
     OUT  (4), A
```

(2) Issue ALARM (WAKE) SET to the 7508 (to set the alarm/wake time).

(3) Set up the work areas.

- Load ALRMTP with 01H.
- Load ALRMMSG (0F3FFH for overseas version and 0F174H for Japanese-language version) with an alarm message in the following format:


41 bytes
4Ø bytes
(Actual message text is loaded here)
Message length (Ø - 4Ø) in binary. A Ø indicates that no message text is defined.

- Load ALRMST with ØØH.

(4) Issue ALARM (WAKE) ON to the 7508 (to enable alarm/wake interrupt).

(5) Enables interrupts from the 7508.

```
     LD   A, (IER)
     OUT  (4), A
```

Take steps (1) through (5) in sequence.

2) Wakel setting procedure

(1) Take the same steps as in alarm setting procedure except step 3):

(3) Set up the work areas.

- Load ALRMTP with Ø2H.
- Load ALRMMSG with a wake string in an appropriate format (a control code is counted as one byte).
- Load ALRMST with ØØH.

3) Wake2 setting procedure

(1) Take the same steps as in alarm setting procedure except step 3):

(3) Set up the work areas.

- Load ALRMTP with Ø3H.
- Load ALRMAD with the address of the processing routine to be executed when a wake interrupt occurs.
- Load ALRMST with ØØH.

8.8 Relationship to BIOS

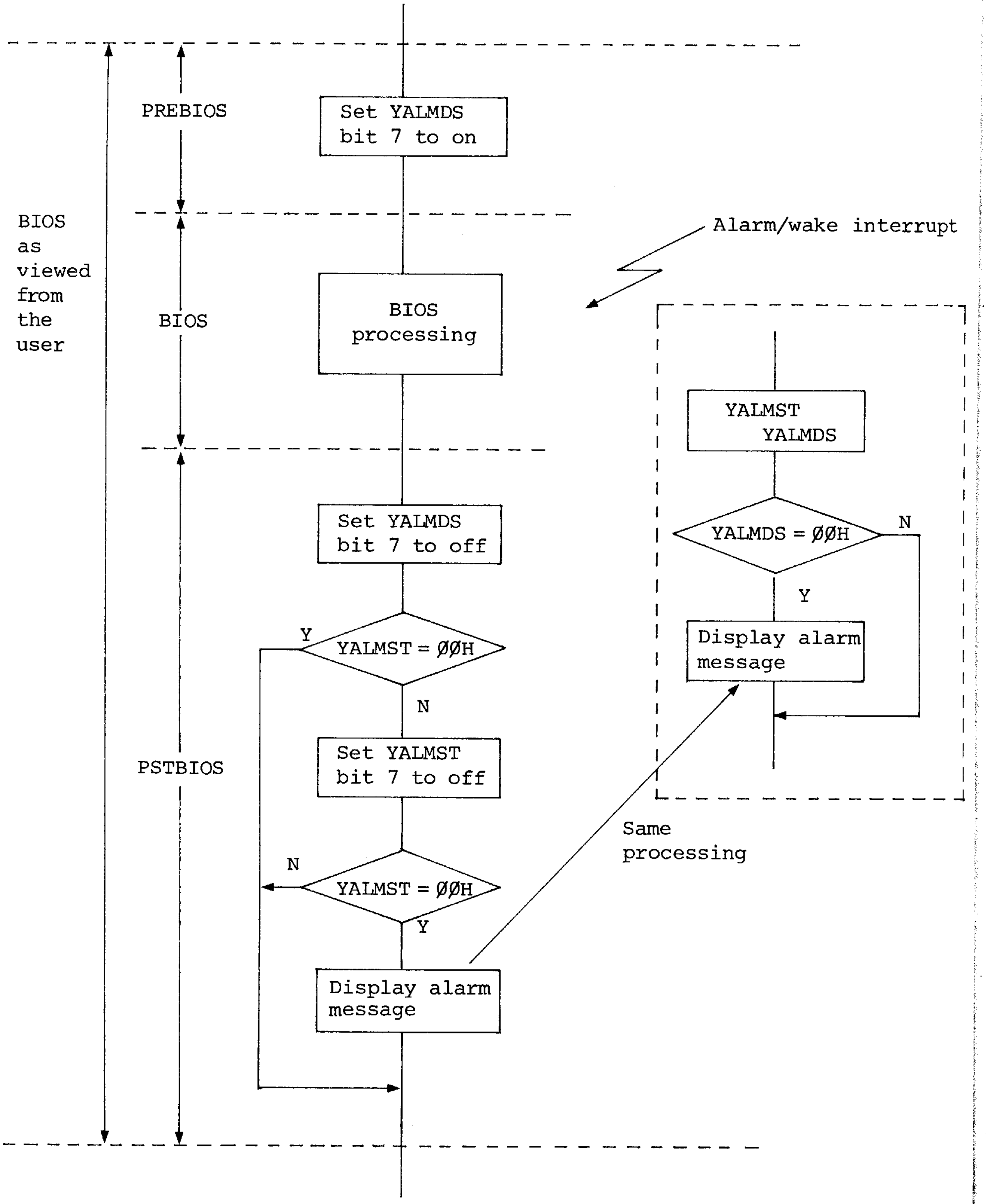
Normal alarm processing displays an alarm message immediately when an alarm interrupt occurs.
When displaying the alarm/wake message, it uses the speaker and screen which are controlled by the slave CPU (63Ø1). The slave CPU does many I/O operations in addition to alarm/wake operation. If an alarm/wake interrupt occurs while the slave CPU is performing an I/O operation and the associated interrupt processing routine attempts to use the slave CPU, the alarm/wake operation overlaps the executing operation, causing a system hang-up.

The MAPLE takes the following measure to solve this problem:
Since the slave CPU runs only when BIOS is performing an I/O operation, the BIOS preprocessing routine sets a flag on and the BIOS postprocessing routine resets that flag. During the time this flag is on, the alarm/wake interrupt processing routine displays no alarm message when an alarm/wake interrupt occurs. It does nothing but flags to indicate that an interrupt has occurred. The BIOS postprocessing routine examines this flag to see whether an alarm/wake interrupt has occurred and

displays an alarm/wake message if the flag is on. The BIOS preprocessing and postprocessing routines are called PREBIOS and PSTBIOS, respectively.

The flag indicating whether the alarm/wake message is to be displayed or not is stored in the YALMDS work area. The flag indicating that an alarm/wake interrupt has occurred while the display of the alarm/wake message is disabled is stored in the YALMST work area.

The figure on the next page shows the relationship between PREBIOS, PSTBIOS, and BIOS, and the relationship of YALMDS and YALMST to BIOS.

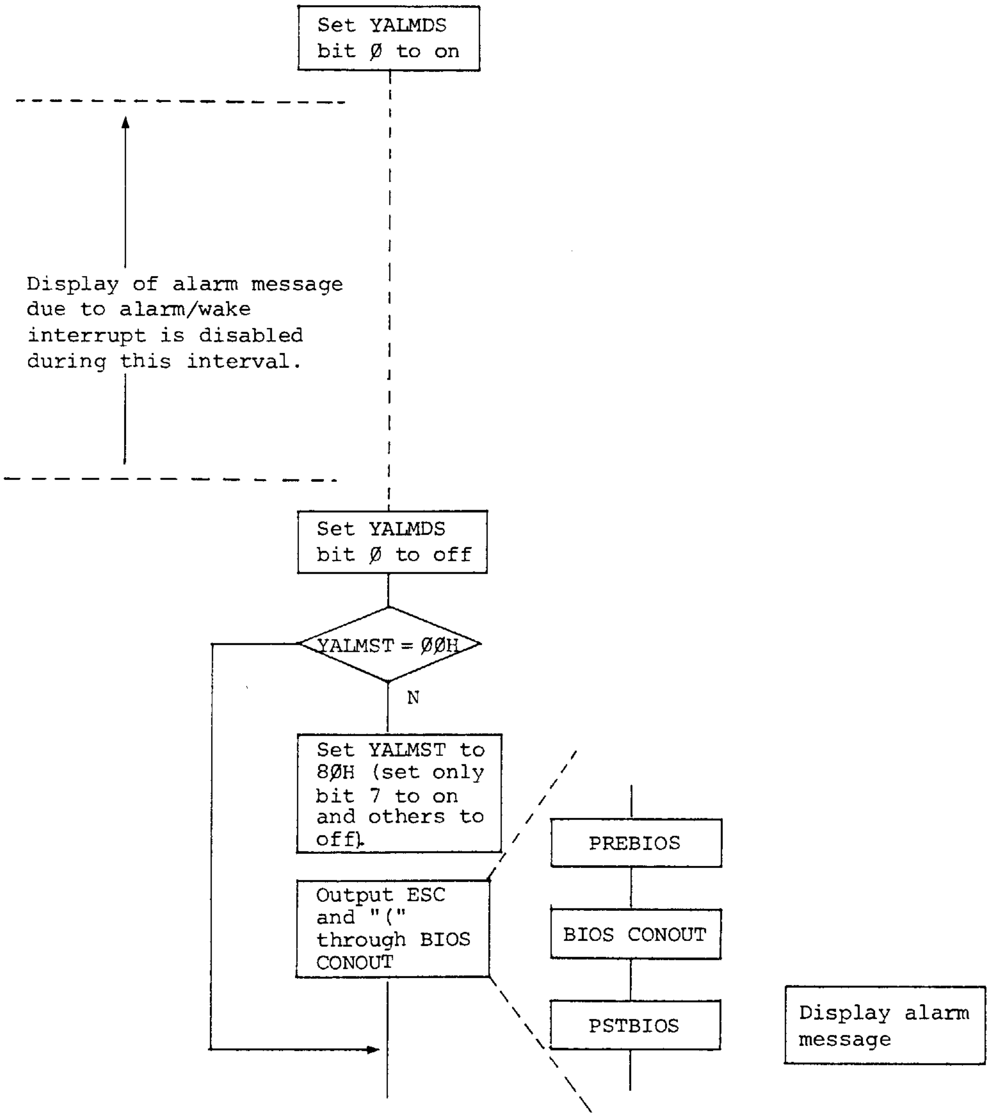

BIOS
as
viewed
from
the
user
PREBIOS
Set YALMDS bit 7 to on
Alarm/wake interrupt
BIOS
BIOS processing
YALMST
YALMDS
YALMDS = ØØH
N
Y
Display alarm message
PSTBIOS
Set YALMDS bit 7 to off
Y
YALMST = ØØH
N
Set YALMST bit 7 to off
N
YALMST = ØØH
Y
Display alarm message
Same processing

8.9 Method of Inhibiting Alarm Message Display from Application Program

Some application programs may not want the alarm message to be displayed during execution of some specific operations. The alarm operation can be disabled by executing the DI instruction or by inhibiting interrupts from the 7508 CPU. These measures, however, will also inhibit other interrupts (e.g., keyboard and power switch off interrupts). To avoid this, the application program can and must perform the same operations as PREBIOS and PSTBIOS do as explained in 8.8.

Display of the alarm message can be disabled using the YALMDS work area. As explained on page 8-18, YALMDS specifies what mode inhibits the alarm/wake interrupt. Bit 0 of YALMDS is reserved for application programs. The procedure on the next page shows how to inhibit alarm/wake operation from the application program.

When an alarm/wake interrupt occurs while alarm message display is disabled (YALMST contains a nonzero value), set only YALMST bit 7 to on (to make the system believe that the interrupt occurred during BIOS processing) and make a dummy call to BIOS. PSTBIOS will then check YALMST bit 7 and display an alarm message.


Set YALMDS bit Ø to on
Display of alarm message due to alarm/wake interrupt is disabled during this interval.
Set YALMDS bit Ø to off
YALMST = ØØH
N
Set YALMST to 8ØH (set only bit 7 to on and others to off)
Output ESC and "(" through BIOS CONOUT
PREBIOS
BIOS CONOUT
PSTBIOS
Display alarm message

The BIOS CONOUT function only passes ESC and "(" ,but no operation results.

8.10 How to Disable System Display Function for Displaying Alarm/Wake Message

When an application program, e.g., scheduler, controls alarm/wake function, malfunctions will result if alarm/wake is set or reset from System Display. To avoid this, the MAPLE OS provides a work area for inhibiting the control of the alarm/wake functions through the System Display.

ALRMDS: Overseas version = 0F06BH
Japanese-language version = 0ED6CH
= 00H enables the control of the alarm/wake functions through the system display function.
≒ 00H disables the control of the alarm/wake functions through the system display function.

ALRMDS defaults to 00H.
ALRMDS is set to 00H by a system initialize.

8.11 Precautions on the Use of the Alarm/Wake Functions

(1) An alarm/wake interrupts are deferred up to 10 seconds in the power-off state. This is because the system checks the alarm/wake time only once every 10 seconds when the MAPLE is in the power-off state.

(2) Since display of the alarm message is inhibited while an BIOS operation is in progress as explained in 8.8, display of the alarm message will be put off accordingly. This should normally be negligible; however, it will be in the order of seconds if the MCT is running.

(3) The Overseas Version B allows the user to change the interval during which the alarm message is displayed (default is 50 seconds) in the range from 1 to 255 seconds.

ALRMPROD (0F2F9H): Load a number from 1-255. Do not specify 0 because the value 0 is interpreted as 0 second or 256 seconds.