Step 1: Execute a DI instruction.
Disables interrupts during bank switching.
Step 2: Set current bank variable CURBNK to new bank.
Update the current bank information in CURBNK (ØF534H) with new bank information.
Step 3: Prepare data for BANKR.
Load system memory area RZBANKR (ØF53DH) for BANKR (PØ5H).
RZBANKR and BANKR contain the same data.
Step 4: Write data to BANKR.
Output new bank data into BANKR (PØ5H). The bank data must be identical to that set up in step 3.
Step 5: Execute an EI instruction.
Enables the CPU for interrupts.

4.4.4 Work Areas Associated with Bank Switching

OLDBNK (ØF52CH) 1 byte
Save area loaded with bank information on entry to BIOS.
= ØFFH: System bank
= ØØH: Bank Ø (RAM)
= Ø1H: Bank 1
= Ø2H: Bank 2

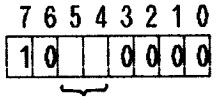
DISBNK (ØF52EH) 1 byte
Parameter area for JUMP or CALL bank switching routine.
= ØFFH: System bank
= ØØH: Bank Ø (RAM)
= Ø1H: Bank 1
= Ø2H: Bank 2

BNKRGS (ØF53ØH) 1 byte
System bank area. Set to ØØH.

BNKRG1 (ØF531H) 1 byte
Bank Ø area. Set to 4ØH.

RZBANKR (ØF53DH) 1 byte
System area for holding output data to RANKR (PØ5H).
Has the same format as BANKR.

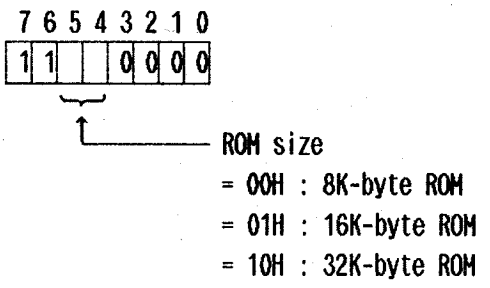
BNKRG2 (ØF532H) 1 byte
Bank 1 area



ROM size
= 00H : 8K-byte ROM
= 01H : 16K-byte ROM
= 10H : 32K-byte ROM

8K-byte ROM is assumed when no ROM is installed.

BNKRG3 (ØF533H) 1 byte
Bank 2 area



8K-byte ROM is assumed when no ROM is installed.

CVRBNK (ØF534H) 1 byte
Contains the bank number of the current bank.
= ØFFH: System bank
= ØØH: Bank Ø (RAM)
= Ø1H: Bank 1
= Ø2H: Bank 2

4.5 Resident Processing

4.5.1 General

When resident processing is specified by an application program, the current program in RAM is preserved and when a power-on condition is generated in the restart mode (by the wake function or other factor), control is transferred immediately to address 1ØØH on bank Ø.

This function allows the PINE to warm-start an application program at a power-on time without passing through the menu processing or CCP, making it a turn-key system.

When this function is activated, the auto start or wake string is passed to the application program through the keyboard buffer. The application program can treat the string as if it were entered from the keyboard at warm-start time.

4.5.2 How to Specify and Cancel Resident

4.5.2.1 Specifying resident

Resident must be specified within the application program. The method of specifying resident processing differs between ROM-based and load-and-go programs.

Resident can be specified by setting the resident flag RESEXQ (ØEF28H) in the system area in one of the following ways:
1. Calling the BIOS RESIDENT function (WBOOT + 84H).
2. Writing Ø1H directly into RESEXQ (ØEF28H).

(1) Actions required of a load-and-go program for resident processing
A load-and-go program, when started, is loaded into RAM (bank Ø) at address 1ØØH, after which control is transferred to address 1ØØH. To specify resident processing, the load-and-go program must take the following steps:

1. Set the resident flag.
2. If necessary, rewrite the instruction at address 1ØØH to a jump to the point of warm start.

Subsequently, when power-on, reset, or WBOOT processing is initiated, the system transfers control directly to address 1ØØH in RAM.

(2) Actions required of a ROM-based program for resident processing
When a ROM-based program is started, the first sector of the program is loaded into RAM (bank Ø) at address 1ØØH, after which control is transferred to the specified program start address. The ROM-based program which is to specify resident processing must load necessary data in advance into the first 9Ø bytes of the ROM program in the specified format (see Section 4.61, "Executing a ROM Program" for the format of the 9Ø-byte data). The ROM-based program need only set the resident flag.

Subsequently, when power-on, reset, or WBOOT processing is initiated, the system transfers control directly to address 1ØØH in RAM. The 9Ø bytes from address 1ØØH contains the data that was read when the ROM-based program was started for the first time. The data consists of routines for checking for a program for which resident is specified, for transferring control to the given address, and for handling errors.

4.5.2.2 Canceling resident processing

Resident processing must be canceled by the application program. Resident can be canceled by:

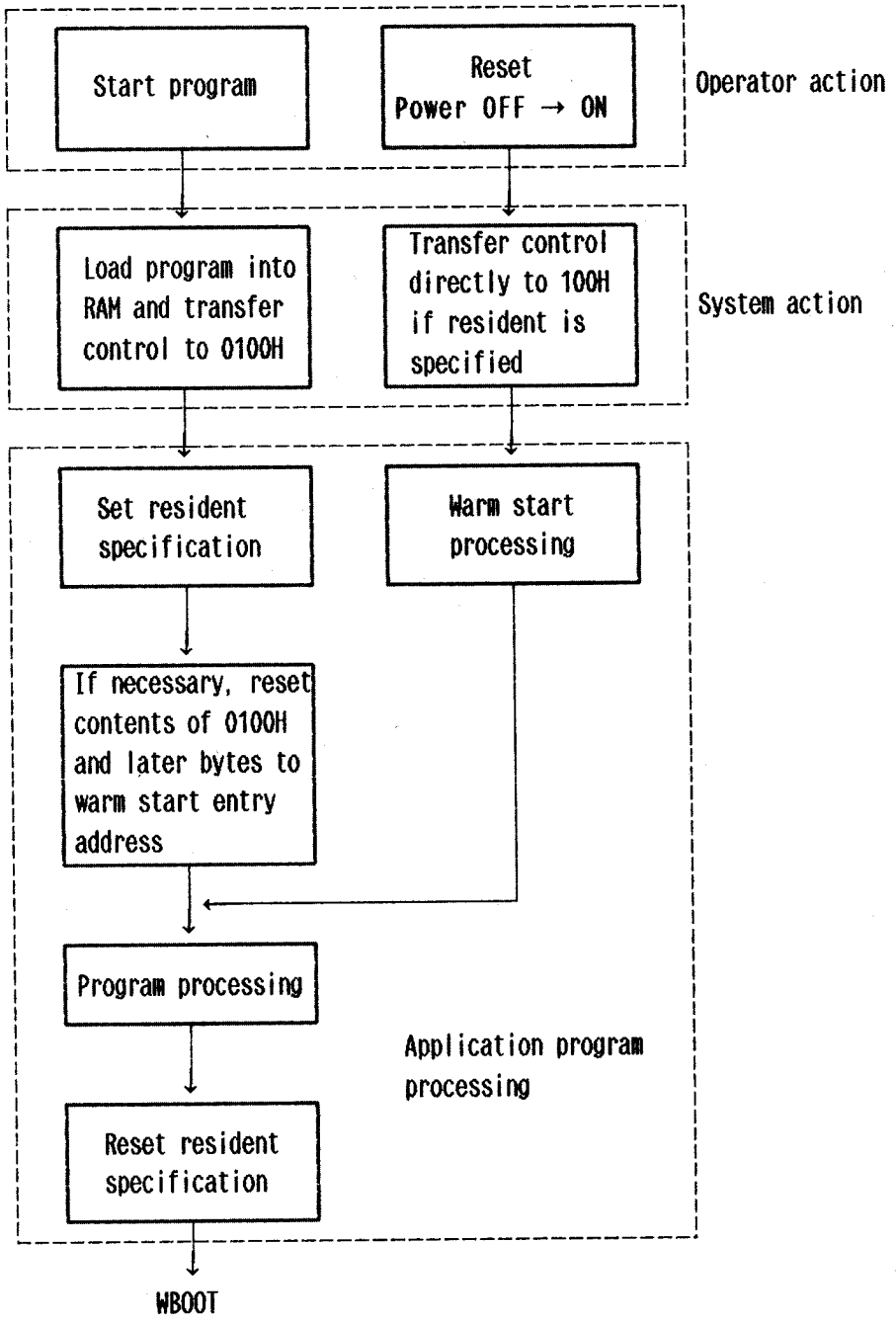
1. Calling the BIOS RESIDENT function (WBOOT + 84H).
2. Writing ØØH directly into RESEXQ (ØEF28H).

The resident flag RESEXQ is not affected by resets; it is cleared to zero only by System Initialize. This means that control cannot exit the application program if the resident flag remains set.

4.5.3 Processing Flows

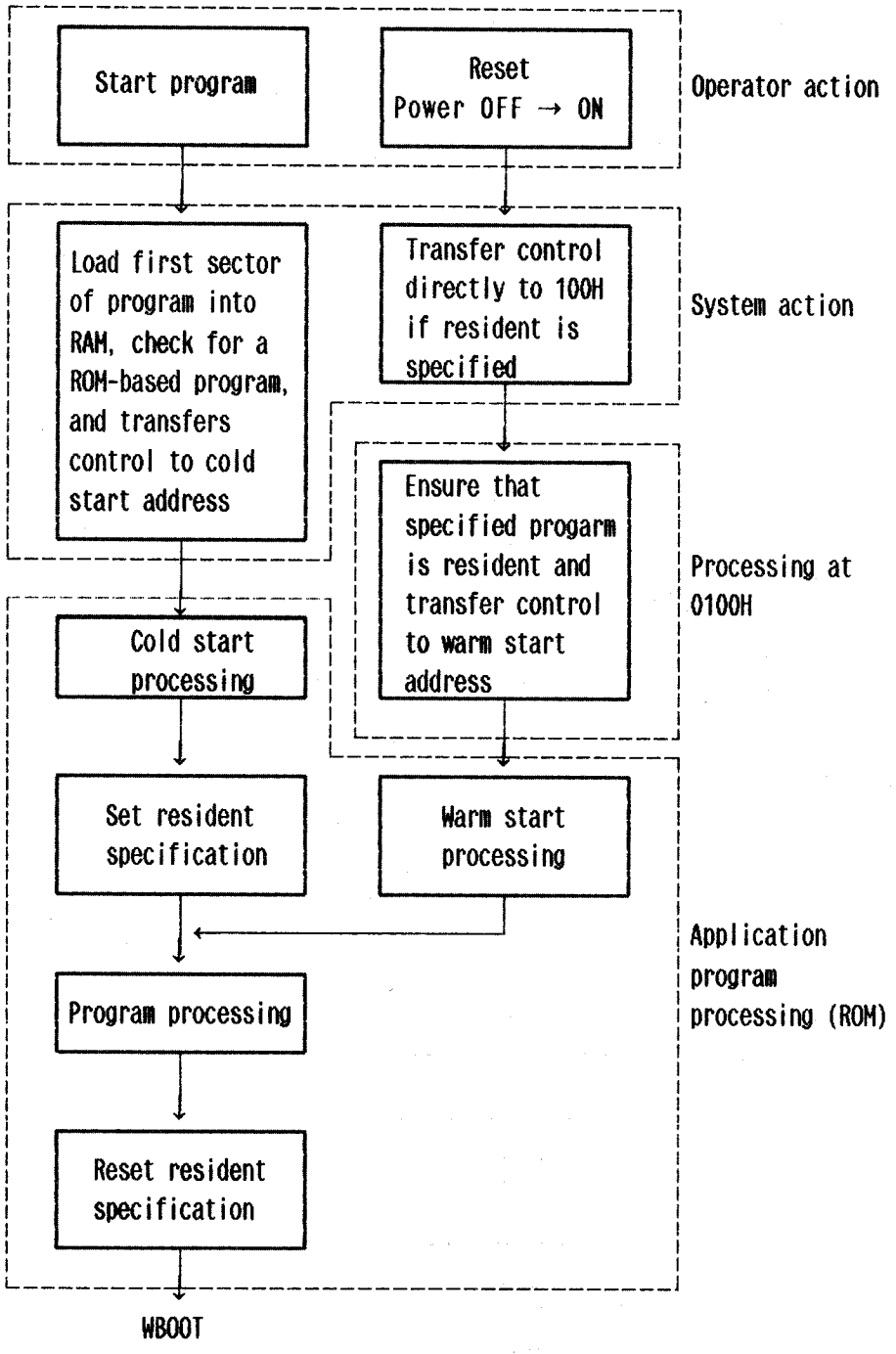
The flowcharts below show the flows of processing of the system and application programs for which resident is specified.

(1) Load-and-go program



(2) ROM-based program

ROM-based programs to be executed with resident specified must begin with a 90-byte data area in the specified format. See Section 4.6, "Executing a ROM Program" for the 90-byte data format.



4.5.4 Miscellaneous Considerations

4.5.4.1 Programming notes

(1) Before terminating (calling WBOOT) a program for which resident is specified, cancel the resident specification. Otherwise, the program can be terminated by no means but system initialize.
(2) When specifying resident processing for a ROM-based program, be sure to define predetermined data in the first 90-byte are of the program in advance. Otherwise, warm start processing will fail.

4.5.4.2 Related Work Area

RESEXQ (0EF28H) 1 byte
Resident flag.
= 00H: Resident not specified.
= Nonzero: Resident specified.
The initial value is 00H. This area is initialized only by system initialize.

4.6 Executing a ROM-based Program

4.6.1 General

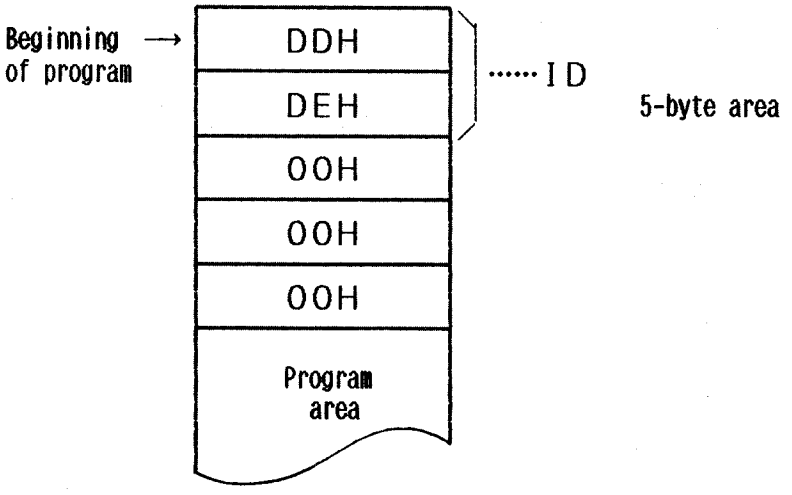
Although the PINE can load application programs from ROM capsules into RAM memory at address 100H for execution as under standard CP/M, it can also execute the programs as stored in the ROM to increase memory efficiency and to save electric power. This section explains the principle of ROM-based program execution, method of establishing the execution environment, and programming notes.

4.6.2 Establishing the Execution Environment

ROM-based programs, which are executed immediately in a ROM capsule, must be given an identification at their beginning to distinguish themselves from ROM programs, called load-and-go ROM programs, which are loaded into RAM memory for execution.

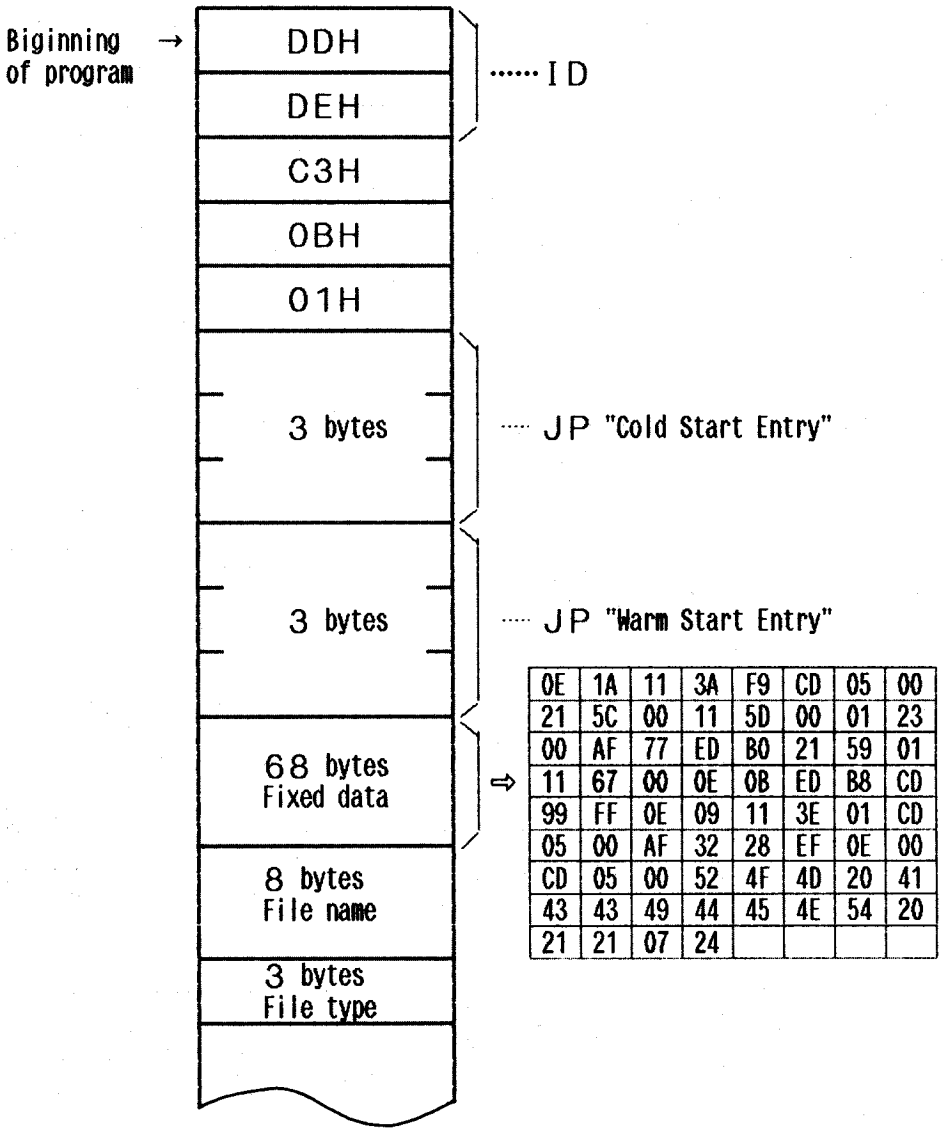
4.6.2.1 When resident is not specified

A ROM-based program without resident specification must begin with a 5-byte ID data as shown in the figure below. The system identifies a ROM-based program by examining this field.



4.6.2.2 When resident is specified

A ROM-based program with resident specified must begin with 90 bytes of information in the format shown below.



0E	1A	11	3A	F9	CD	05	00
21	5C	00	11	5D	00	01	23
00	AF	77	ED	B0	21	59	01
11	67	00	0E	0B	ED	B8	CD
99	FF	0E	09	11	3E	01	CD
05	00	AF	32	28	EF	0E	00
CD	05	00	52	4F	4D	20	41
43	43	49	44	45	4E	54	20
21	21	07	24				

Offset	Bytes	Name	Description
ØØH~Ø1H	2	ID	ROM-based program identifier
Ø2H~Ø4H	3	Jump instruction for resident processing	Jump instruction for resident processing (NOP when resident is not specified)
Ø5H~Ø7H	3	Cold Start Entry	Program cold start address (Jump to ROM cold start address)
Ø8H~ØAH	3	Warm Start Entry	Program hot start address (Jump to ROM hot start address)
ØBH~4EH	68	Resident processing	Instructions for resident processing. Resident processing consists of: 1. Setting up DMA and FCB. 2. Executing RSROMEXQ. 3. Abnormal termination processing.
4FH~56H	8	File name	ROM-based program name (must be the same as that in the ROM directory)
57H~59H	3	FILE TYPE	ROM-based program type (must be the same as that in the ROM directory)

4.6.3 Processing Flow

Figure 4.6.1 shows the flow of processing up to the point where a ROM-based program receives control.

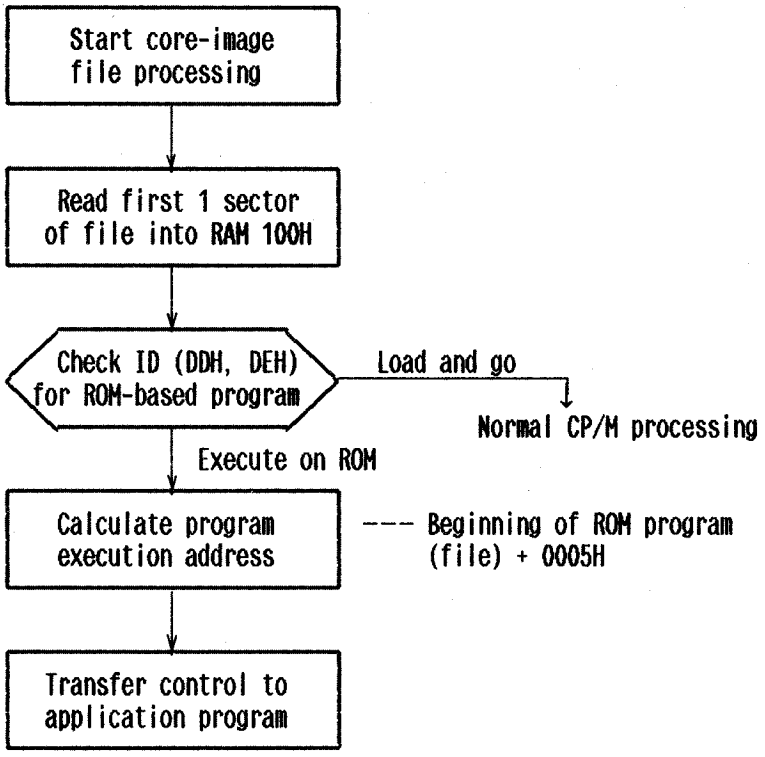


Fig. 4.6.1 Starting a ROM-based Program

4.6.4 Use and Programming Notes

This subsection describes the use and programming notes to be taken when programming ROM-based programs.

4.6.4.1 Program starting address

Load-and-go ROM programs are loaded into RAM memory at location 1ØØH for execution. Such programs must be allocated RAM memory starting at address 1ØØH. ROM-based programs can execute directly in a ROM capsule. The starting address of such programs must be determined by taking into account the header and directory areas.

The program starting address differs depending on the size of the ROM-based program and the type of the PROM (27256, 27128, or 2764) in which it is to be stored.

A ROM-capsule must contain header and directory areas. Since the size of the directory area of a program file varies with the number and size of the programs that the file contains (though its header area is fixed at 32 bytes), it is necessary to calculate the size of the directory area to be reserved before actually loading the programs.

Figure 4.6.2 shows the relationship between the ROMs and PINE addresses.

Note that the physical addresses of 27128 and 27256 ROMs do not match the PINE logical addresses.

When programming a ROM program, use the optional PROM WRITEr cartridge or PROMFORM utility program.

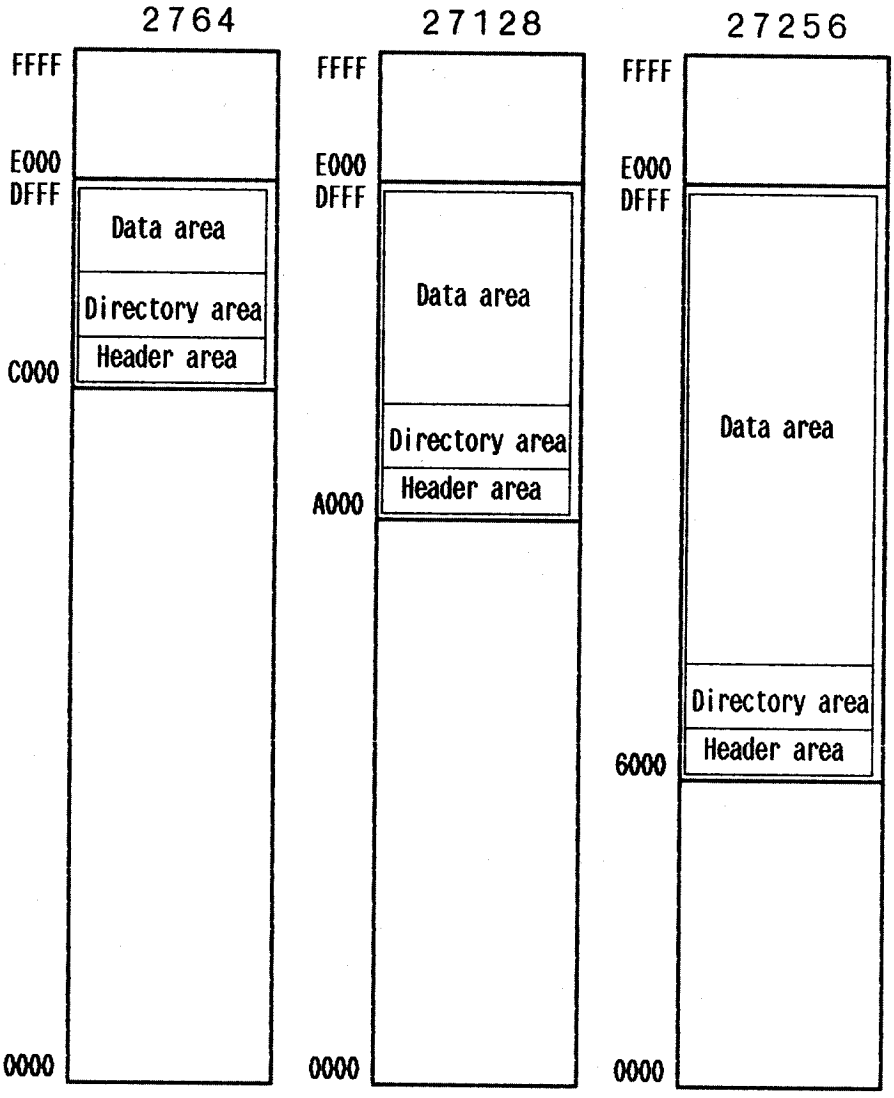


Fig. 4.6.2 Relationship between Logical and ROM Addresses
(ROM area is enclosed in double boxes)

(1) When implementing a single ROM-based program in a ROM
The program starting address of a ROM-based program consisting of a single program is listed below.

ROM	2764	27128	27256
Starting address	0C080H	0A080H	6080H

A ROM-based program of this type must begin with a 32-byte header and a 32 bytes x 3 directory area. Accordingly, the program starting address must be 80H bytes away from the beginning of the ROM. See Section 7.3, "Guide for Programming ROMs" for the structure of the header and directory areas.

(2) When implementing more than one ROM-based program in a ROM
When implementing more than one ROM-based program in a ROM, it is necessary to calculate in advance the size of the directory area to be reserved since the size of a portion of the directory area varies with the number of programs and ROM capacity.

One directory entry can handle a file of up to 16K bytes. Two or more entries are used to handle files larger than 16K bytes. For example, two directory entries are required for a ROM-based program larger than 16K bytes and one directory entry for one which is smaller than 16K bytes.

Since the directory area (including the header area) is reserved in 128-byte increments, the size of space necessary for the header and directory areas is obtained from the formula:

$$[\{((n1/16384 + 1) + (n2/16384 + 1) + \ldots)/4\} \times 4 + 1] \times 32 \text{ bytes}$$

Number of directory entries

Total number of directory entries

For 128-byte boundary alignment

(16K = 16,384)

where n1, n2, ... denote program sizes.

The program starting address is obtained by adding the above result to the starting address of the ROM.

To find the starting address of the second and subsequent programs, follow the steps given below.

Since CP/M manages files in 1K-byte units, the starting address of the second or subsequent program is obtained by adding the starting address of the preceding program plus

$$[\{(n-1)/1024\} + 1] \times 1024 \quad \text{(bytes)}$$

where n is the size of the preceding program.

Figure 4.6.3 shows the memory map for the following ROM programs implemented in a 27256 ROM:

File name	Records	File size	Directory entries	Starting address
FILE1.COM	30	4K bytes	1	6100H
FILE2.COM	150	19K bytes	2	7100H
FILE3.COM	50	7K bytes	1	BD00H

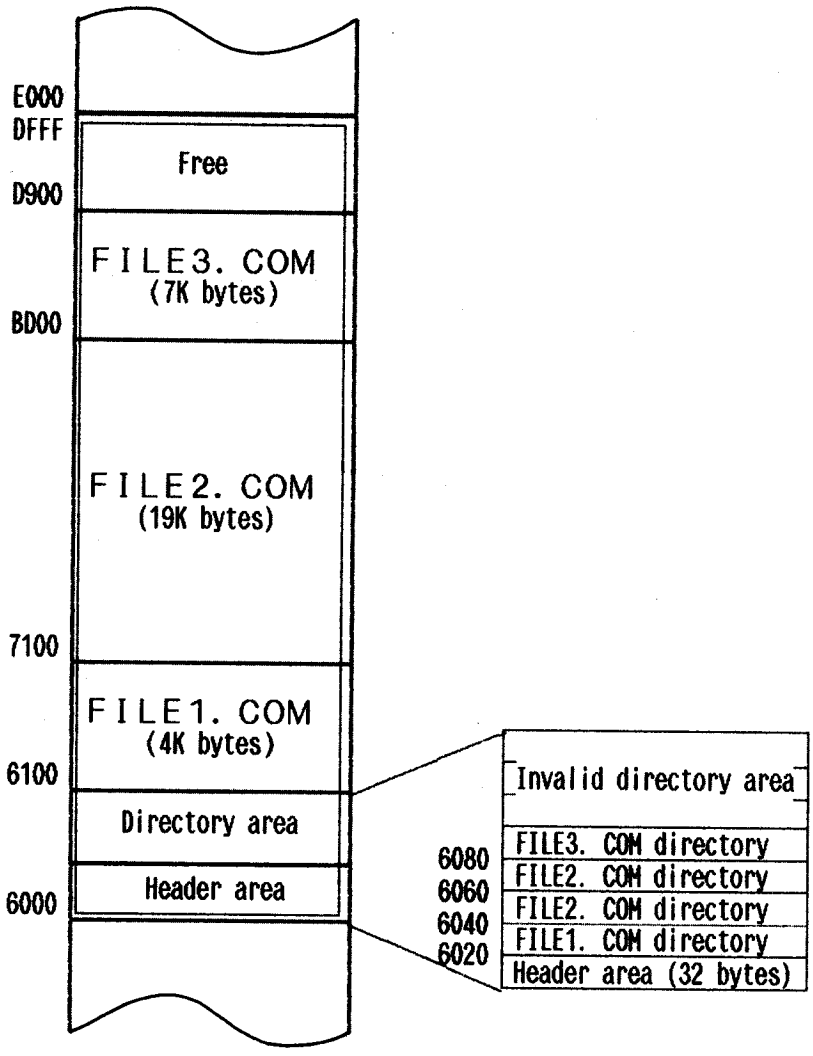


Fig. 4.6.3 Sample ROM Memory Map (for PINE)

(2) ROM-based Program Work Areas

ROM-based programs need a work area which is different from that used by load-and-go programs. Whereas load-and-go programs can use the area following their program area as a work area, ROM-based programs must use an area higher than 100H (15AH or higher when Resident is specified). The upper limit of the work area for ROM-based programs is obtained as follows:

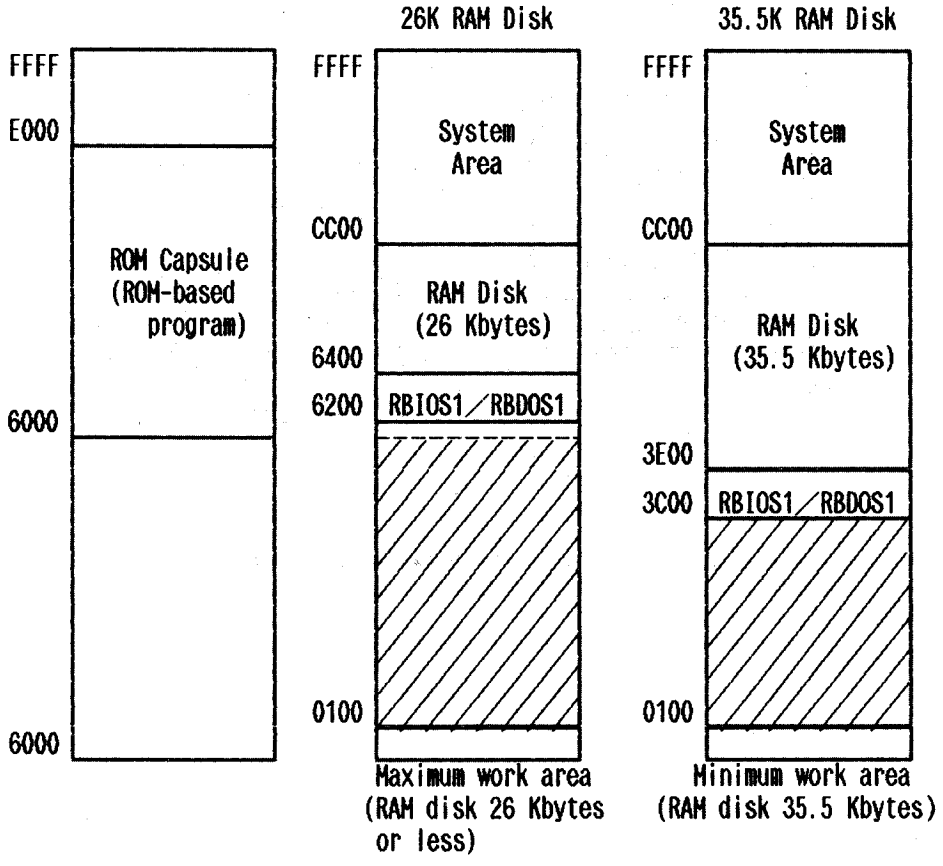
min{(ROM Starting Address), (RBDOS1 Starting Address)}

The ROM starting address is:

- 6000H for 27256
- 0A000H for 27128
- 0C000H for 2764

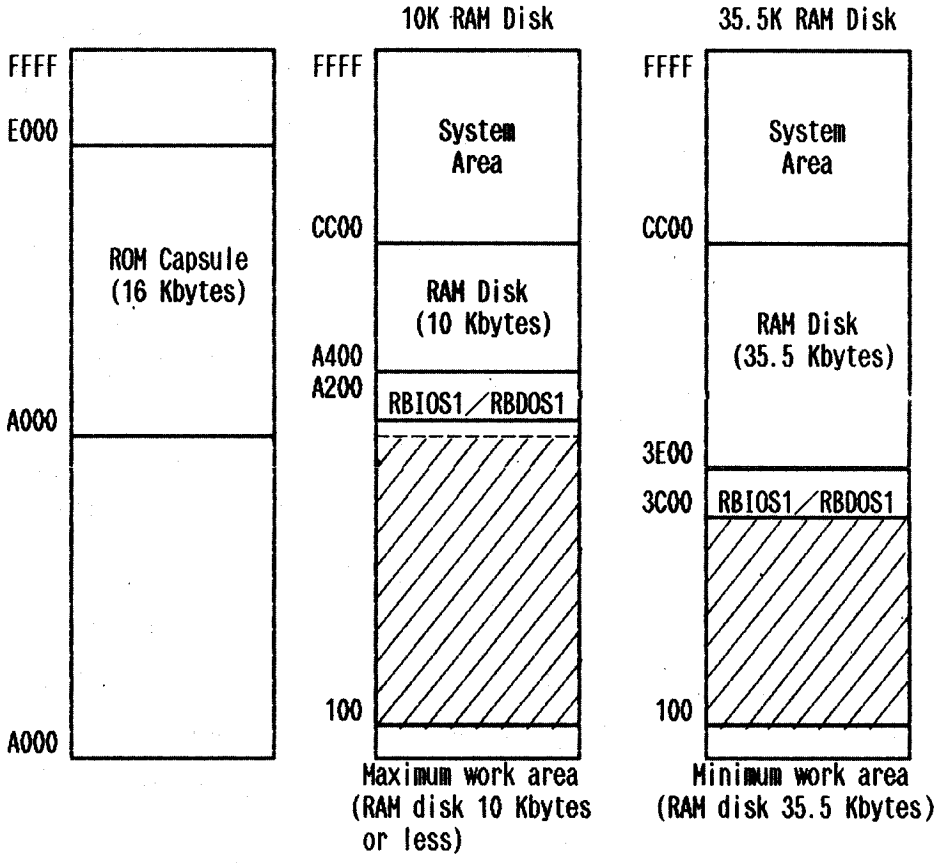
The RBDOS1 staring address is stored in addresses 0006H and 0007H.

a) When 27256 ROM is used



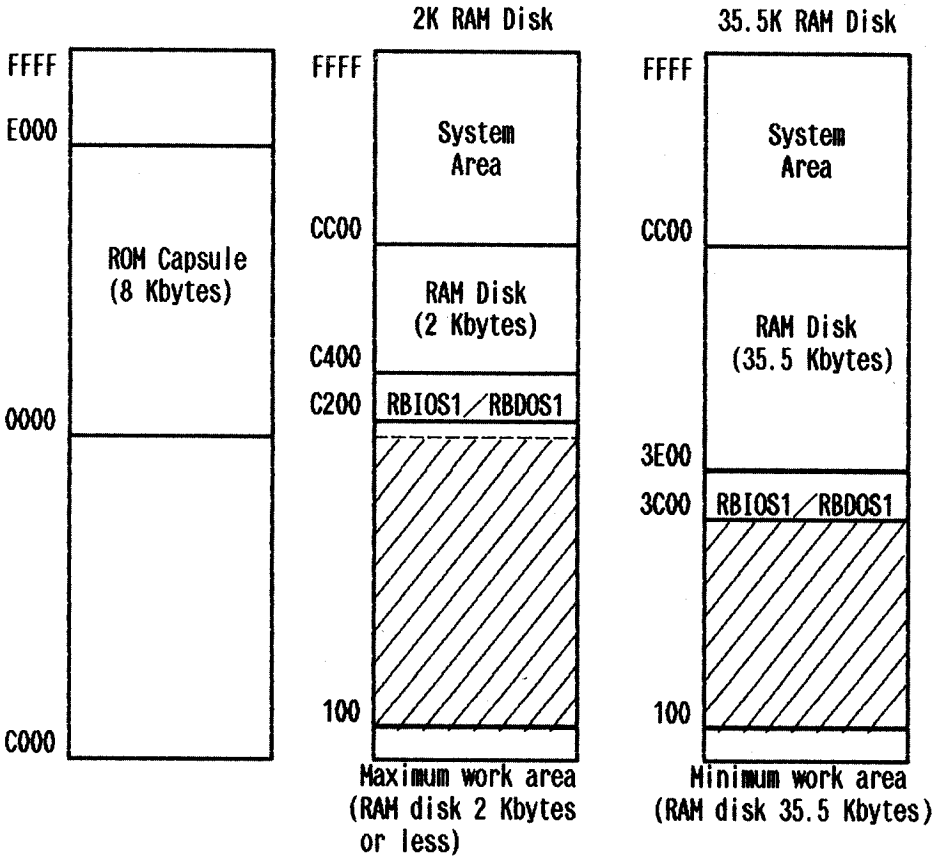
The hatched area is available as the work area without bank switching. The area starting at 6000H is made available through bank switching.

b) When 27128 ROM is used



The hatched area is available as the work area without bank switching. The area starting at address ØA000H is made available through bank switching.

c) When 2764 ROM is used



The hatched area is available as the work area without bank switching. The area starting at address ØCØØØH is made available through bank switching.

(3) BDOS/BIOS Calls from ROM-based Programs
Load-and-go programs call address 0005H through 0007H when making a BDOS call and address 0000H through 0002H when making a BIOS call. For ROM-based programs, however, the addresses of BDOS and BIOS may be located in the background memory of the ROM capsule (because their location varies with the capacity of the installed RAM disk). When calling BDOS and BIOS, therefore, ROM-based programs must call BDOS and BIOS entry points (RBDOS2 and RBIOS2) in the resident area.

```
RBDOS2 entry address:     0FF90H     (JP RBDOS2)

RBIOS2 BOOT address:       0EB00H
RBIOS2 WBOOT address:      0EB03H
RBIOS2 CONST address:      0EB06H

          |                  |

RBIOS2 CONTINUE address: 0EB8AH
```

(4) ROM-based program ROMs
The PINE supports two formats, P- and M-formats, for ROM capsules. ROM-based programs must be programmed in the P-format. Load-and-go ROM programs may be programmed either in the P- or M-format. See Section 7.3, "Guide for Programming ROMs" for instructions for programming ROMs.

4.7 Interrupts

4.7.1 General

PINE supports five types of interrupts. When an interrupt occurs, it takes in and sets up the corresponding interrupt vector and transfers control to the interrupt processing routine.

The interrupts that the PINE supports include:
1. 7508 (4-bit CPU)
2. ART (RxRDY)
3. Alarm time
4. ICF (Input Capture)
5. OVF (FRC Overflow)
6. EXT (External)

The sources of 7508 interrupts are:
1. Keyboard
2. Power switch
3. Alarm time
4. Power fail
5. 1-second interrupt

4.7.2 Interrupt Vector

The PINE handles five types of interrupts. Their interrupt vector table is located in the area between 0FFF0H and 0FFFFH.

Interrupts are given priorities; the PINE accepts interrupts of the highest priority first.

Table 7.4.1 Interrupt Vector Table

Priority	Interrupt source	Vector	RAM address
1 (highest)	7508 (4-bit CPU)	0F0H	0FFF0H and 0FFF1H
2	ART (RxRDY)	0F2H	0FFF2H and 0FFF3H
3	ICF (Input Capture)	0F4H	0FFF4H and 0FFF5H
4	OVF (FRC Overflow)	0F6H	0FFF6H and 0FFF7H
5	EXT (External interrupt)	0F8H	0FFF8H and 0FFF9H

The interrupt sources and their reset conditions are listed in Table 4.7.2.

Table 4.7.2 Interrupt Sources and Reset Conditions

Interrupt	Interrupt source	Reset conditions
75Ø8	* Keyboard input * Lapse of 1 second * Alarm time * Power switch on or off * Power fail voltage	Returns a reply to the 75Ø8.
ART (RxRDY)	* ART RxRDY set.	Reads the receive data register ARTDIR (P14H).
ICF	* Change in state of the input signal from barcode reader or cassette drive	Reads the input capture register (PØ3H).
OVF	* FRC (Free Running Counter) overflows. FRC is a 16-bit counter running at a period of 1Ø6.7 ms.	Issues a Reset OVF command (sets CMDR (PØ2H) bit 2 to 1).
EXT	* Interrupt signal from external device (via system bus)	Returns a reply to the external device (via system bus).

4.7.3 Interrupt Control

The PINE takes various actions to control interrupts.

4.7.3.1 Setting up the interrupt mode and vector data

When an address ØØØØH start occurs (started by a system initialize, reset, or power-on), the PINE specifies the mode 2 interrupt and loads the I register with ØFFH (the higher order 2 digits of the interrupt vector table address).

4.7.3.2 Loading the interrupt vector table

The PINE loads interrupt data from OS ROM into addresses ØFFFØH through ØFFFFH when a reset or system initialize occurs.

4.7.3.3 Interrupt control (disabling and enabling)

(1) Interrupt control by the system
The PINE system disables and enables interrupts as follows:

Table 4.7.3 Interrupt Control

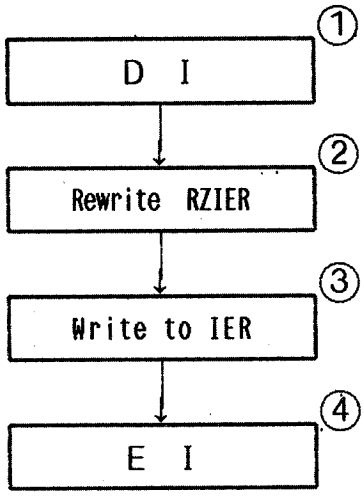
Interrupt / Processing	7508	7508 key	7508 1sec	7508 Alarm	ART	OVF	ICF	EXT	Remarks
System Initialize	○	○	○	×	×	○	×	×	This state continues.
Reset	○	○	○	—	×	○	×	×	
Restart Pw ON	○	—	—	—	×	○	×	×	
Continue Pw ON	—	—	—	—	—	—	—	—	
7508 interrupt processing	×	—	—	—	—	—	—	—	Enabling the CPU for interrupt of 7508 during alarm screen display
ART interrupt processing	×	—	—	—	×	×	×	×	Continues processing in DI state
OVF interrupt processing	×	—	—	—	×	×	×	×	
ICF interrupt processing	×	—	—	—	×	×	×	×	
EXT interrupt processing	×	—	—	—	×	×	×	×	
MCT processing	×	—	—	—	—	×	—	—	
FDD processing	×	—	—	—	×	×	×	×	Continues processing in DI state.
BEEP processing	—	△*	—	—	—*	×	—*	—*	*

* indicates that the interrupt disable/enable state can be changed during processing.
○··· Enabled ×··· Disabled —··· No change
△··· Disables all interrupts except STOP key interrupts.

(2) Disabling or enabling interrupts
The PINE CPU can be disabled or enabled for interrupts by:
1. Using the BIOS MASKI function.
2. Rewriting the Interrupt Enable Register (IER) (P04H) directly.

See Section 3.4, "BIOS Details" for the use of the BIOS MASKI function.

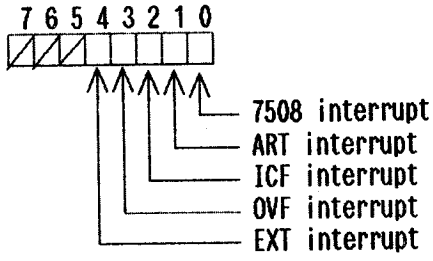
The figure below shows the steps for rewriting the IER directly.




```
RZIER(0F53EH)
  Interrupt state
  (disable/enable)area.
Example : Enabling the CPU
          for external
          interrupts.
          DI
          LD   A,(RZIER)
          OR   10H
          LD   (RZIER), A
          OUT (IER), A
          EI
```

Fig. 4.7.4 Rewriting IER

Related work area:
RZIER (0F53EH) 1 byte
Contains the interrupt disable/enable status.



1 refers to enabling and a 0 to disabling interrupts.
The format of the RZIER is identical to that of the IER.

4.7.3.4 Interrupt processing time

Table 4.7.5 lists the times required for the PINE to process interrupts.

Table 4.7.5 Interrupt Processing Times

Interrupt	Number of machine states	Time	Remarks
7508 (key in)	13789	3747μs	
7508 (1 sec)	14141	3842μs	
7508 (Alarm)	14087	3828μs	Processing time when alarm display processing is disabled
7508 (Pw sw on)	13837	3760μs	
7508 (Pw sw off)	175852	47786μs	Processing time when power-off processing is disabled
7508 (Power fail)	13891	3775μs	Processing time when power-off processing is disabled
ART	1623	441μs	
OVF	1122 (4504)	305μs (1224μs)	Number in parentheses is the processing time during reverse video cursor processing
ICF	686	186μs	
EXT	698	190μs	

Note: The reason that the interrupt processing times during power-off processing are longer than usual is that the PINE waits for approximately 40 ms after issuing a keyboard clear command to the 7508.

4.7.3.5 Interrupt processing

(1) Location where interrupt processing is performed
It is unpredictable on which bank the PINE is running when an interrupt occurs. To keep track of the location of PINE execution when an interrupt occurred, the PINE places the entry portion of the interrupt processing in the resident area (ØE000H through ØFFFFH). When an interrupt occurs, the banks are switched in the entry portion and the actual interrupt processing is carried out in the main interrupt processing routine on the OS ROM.

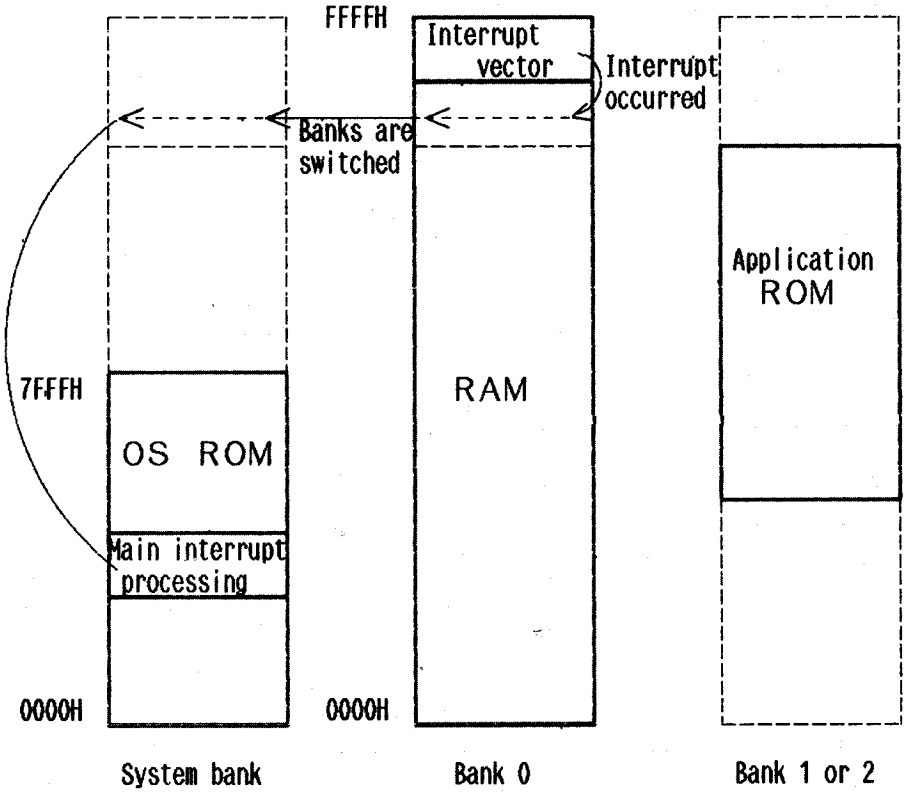


Fig. 4.7.6 Interrupt Processing When an Interrupt Occurred

(2) Relationship between Interrupt processing and BIOS
There are some types of interrupts which, when generated during BIOS processing, prevent the PINE from continuing program execution on return from interrupt processing or from performing successful power-off processing in the continue mode if it processes the interrupt immediately.

To solve this problem, the PINE sets the BIOS in-process flag on entry to BIOS (PREBIOS) and, if an interrupt occurs when this flag is on, causes the interrupt processing routine only to set the interrupt flag to memorize the occurrence of an interrupt. At the end of BIOS processing after the return from the interrupt processing, the PINE tests the interrupt flag and, if it is found to be set, performs the actual interrupt processing at the exit of BIOS (PSTBIOS).

BIOS functions which contain loops (e.g., CONIN, RSIN, and RSOUT) also test the interrupt flag as PSTBIOS does and invoke necessary interrupt processing accordingly.

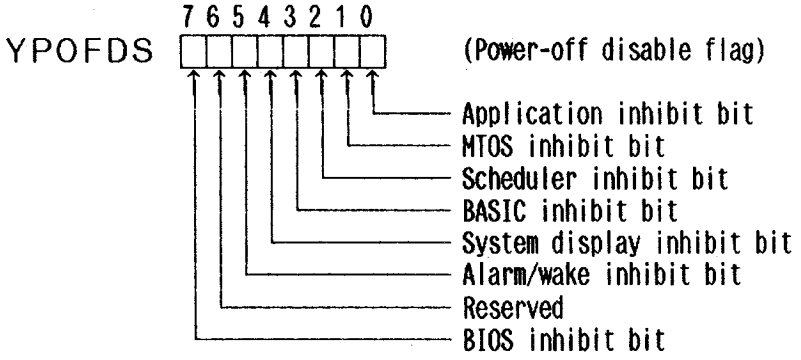
See also Section 3.3, "BIOS Operations" for details on PREBIOS and PSTBIOS.

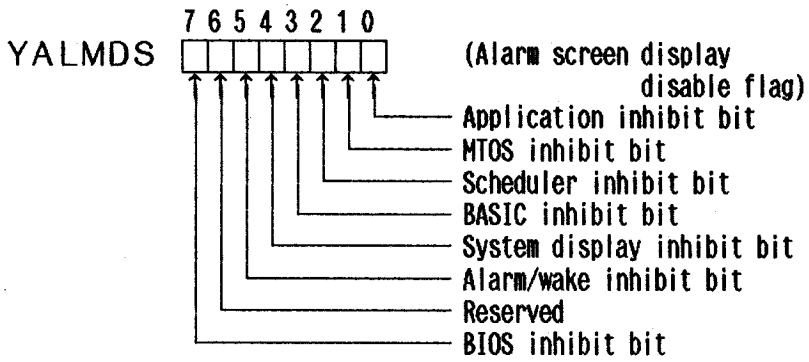
(3) Inhibiting interrupts
The PINE can inhibit the system from performing interrupt processing under program control. Interrupt processing that can be inhibited in this way includes:
1. Power-off processing and power fail processing
2. Alarm/wake processing

"Disabling interrupts under program control" means that the PINE accepts interrupt requests but inhibits the execution of the corresponding interrupt processing.

This facility is used by PREBIOS and PSTBIOS. Application programs can use this facility to temporarily inhibit power-off or alarm processing while they are taking specific actions.

a) How to inhibit interrupt processing
PINE OS uses the YPOFDS (ØEFEFH) and YALMDS (ØEFF1H) flag areas for inhibiting power-off and alarm/wake processing. The format of these areas are shown below.





Setting the corresponding bit to 1 inhibits power-off or alarm screen display processing.

During interrupt processing, the system copies the values of YPOFDS and YALMDS into YPOFST (ØEFFØH) and YALMST (ØEFF2H). If YPOFST or YALMST contains ØØH, the system performs power-off or alarm screen display processing; if both YPOFST and YALMST contains nonzero values, the system does nothing and terminates the interrupt processing.

b) Procedures for inhibiting interrupt processing
Figures 4.7.7 and 4.7.8 show the procedures for inhibiting interrupt processing. Application programs can manipulate only bit Ø of YPOFDS and YALMDS and must not use the other bits.

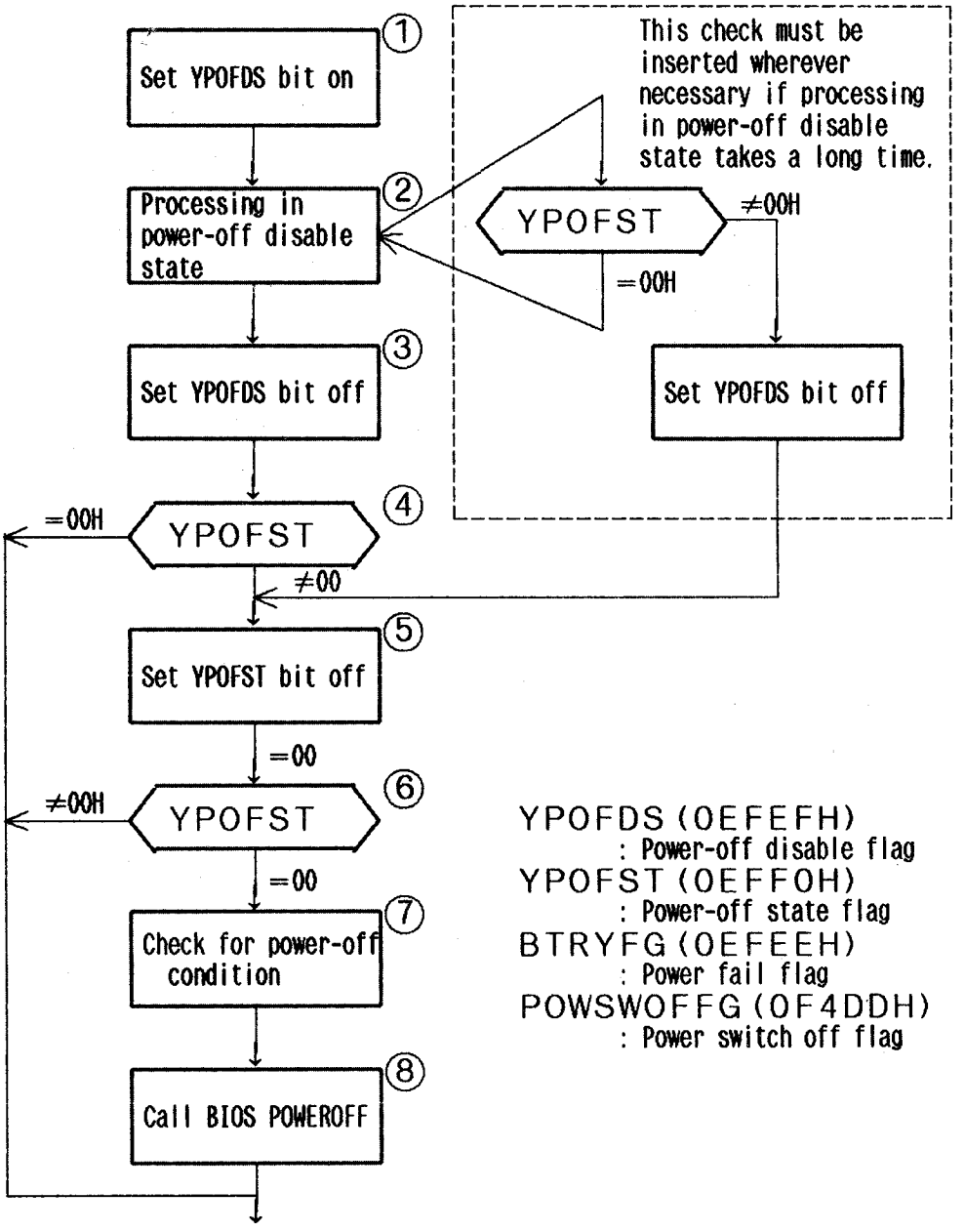


Fig. 4.7.7 Procedure for Inhibiting Power-off Processing

Step 1: Set YPOFDS bit on.
Set the specified YPOFDS (ØEFEFH) bit to 1. The application program must use bit Ø.

Step 2: Processing in power-off disable state.
Perform the processing to be executed in the power-off disabled state.
A power-off check routine must be inserted wherever necessary if the processing in power-off disable state takes a long time.

Step 3: Set YPOFDS bit off.
Set the YPOFDS bit that is set to 1 in step 1 to Ø.

Step 4: Check YPOFST.
Tests the value of YPOFST (ØEFFØH).
A nonzero value in YPOFST indicates that a power-off interrupt occurred while power-off processing was inhibited.

Step 5: Set YPOFST bit off.
Set the YPOFST bit that is set to 1 in step 1 to Ø.

Step 6: Check YPOFST.
Test the value of YPOFST (ØEFFØH).
A nonzero value in YPOFST indicates that power-off processing is inhibited by another module.

Step 7: Check for power-off condition.
Check the power-off conditions.
A nonzero in BTRYFG (ØEFEEH) indicates a continue mode power-off condition. If BTRYFG contains ØØH, the system examines PWSWOFFG (ØF4DDH). A Ø2H in PWSWOFFG indicates the restart condition and the other values indicate the continue-mode power-off condition.

Step 8: Call BIOS POWEROFF.
Call BIOS POWEROFF specifying the power-off condition found in step 7 specified.

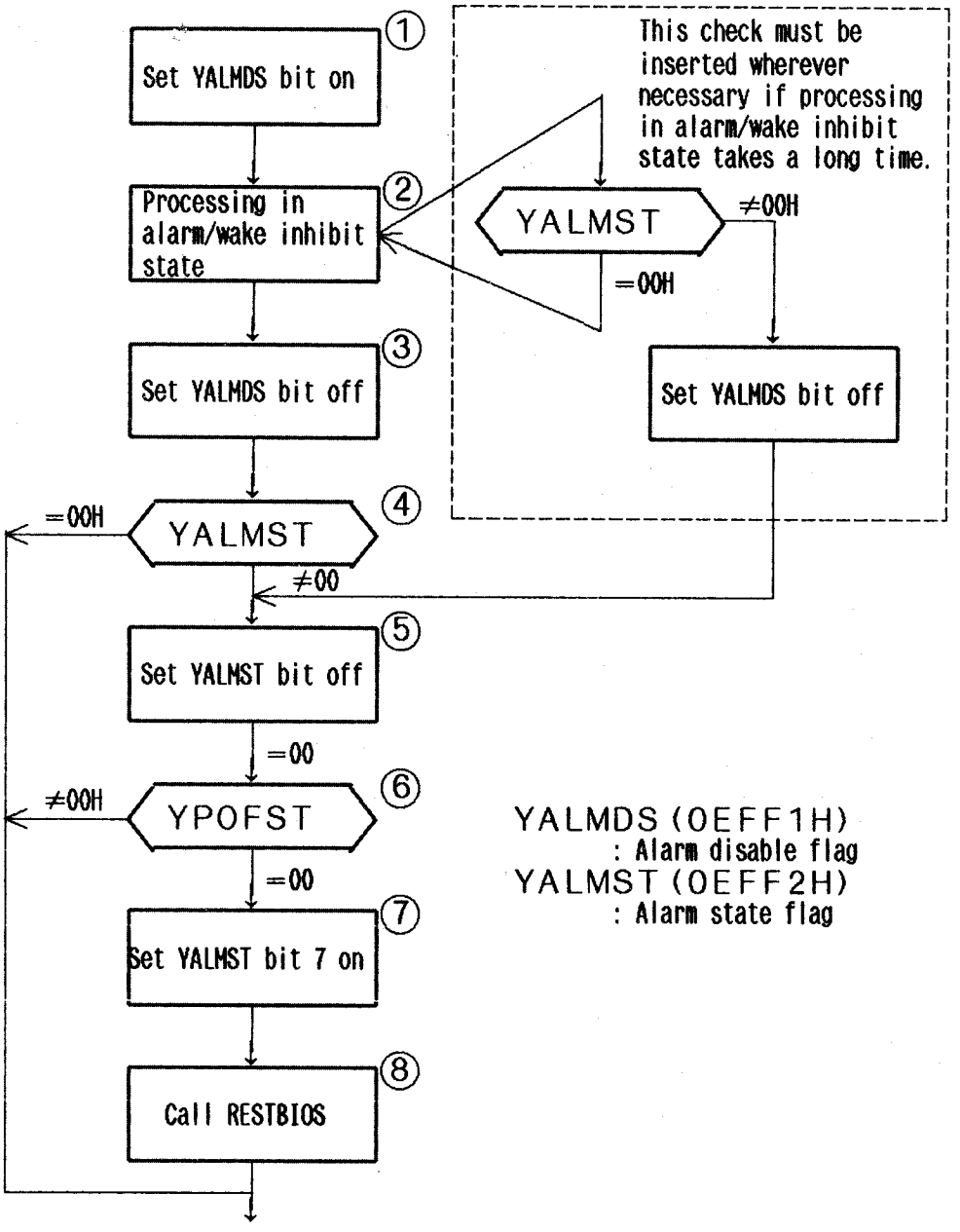


Fig. 4.7.8 Procedure for Inhibiting Alarm Screen Display Processing

Step 1: Set YALMDS bit on.
Set the specified YALMDS (ØEFF1H) bit to 1. The application program must use bit Ø.
Step 2: Processing in alarm/wake inhibit state.
Perform the processing to be executed in the alarm/wake screen display inhibit state.
A alarm/wake check routine must be inserted wherever necessary if the processing in alarm/wake inhibit state takes a long time.
Step 3: Set YALMDS bit off.
Set the YALMDS bit that is set to 1 in step 1 to Ø.
Step 4: Check YALMST.
Tests the value of YALMST (ØEFF2H).
A nonzero value in YALMST indicates that an alarm interrupt occurred while alarm/wake processing was inhibited.
Step 5: Set YALMST bit off.
Set the YALMST bit that is set to 1 in step 1 to Ø.
Step 6: Check YALMST.
Test the value of YALMST (ØEFF2H).
A nonzero value in YALMST indicates that alarm/wake screen display processing is inhibited by another module.
Step 7: Set YALMST bit 7 on.
Set bit 7 of YALMST (ØEFF2H) to 1. This enables alarm/wake screen to be displayed by PSTBIOS.
Step 8: Call RSPSTBIOS.
Call RSPSTBIOS (ØFF96H) to display the alarm/wake screen.

c) Programming notes
1. While the system is waiting for keyboard data with BIOS CONIN, it unconditionally executes power switch off, power fail, and alarm/wake processing even if they are inhibited. This also applies to BIOS which is waiting for send or receive data or for a printer ready signal (except during hardcopy processing).
2. Checks must be made for power-off or alarm/wake conditions if processing in the power-off or alarm/wake processing disabled state is to take a long time.

```
                        ;       ************************************************
                        ;               POWER OFF & ALARM CONTROL PROGRAM
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;               This sample program shows how to control
                        ;               power off & alarm interrupt.
                        ;
                        ;       <> assemble condition <>
                        ;
                        ;       .Z80
                        ;
                        ;       <> loading address <>
                        ;
                        ;       .PHASE  100H
                        ;
                        ;       <> constant values <>
                        ;
                        ;       BIOS entry
                        ;
EB03                    WBOOT           EQU     0EB03H          ; Warm Boot entry
EB06                    CONST           EQU     WBOOT   +03H    ; Console status entry
EB09                    CONIN           EQU     WBOOT   +06H    ; Console in entry
EB0C                    CONOUT          EQU     WBOOT   +09H    ; Console out entry
EB7E                    POWEROFF        EQU     WBOOT   +7BH    ; Power off entry
                        ;
                        ;       System area
                        ;
EFEF                    YPOFDS          EQU     0EFEFH          ; Power off disable flag.
EFF0                    YPOFST          EQU     0EFF0H          ; Power off status.
EFF1                    YALMDS          EQU     0EFF1H          ; Alarm disable flag.
EFF2                    YALMST          EQU     0EFF2H          ; Alarm status.
EFEE                    BTRYFG          EQU     0EFEEH          ; Power fail status.
EFF0                    PWSWOFFG        EQU     0EFF0H          ; Power sw. off status.
                        ;
                        ;         RAM jump table
                        ;
FF96                    RSPSTBIOS       EQU     0FF96H          ; Post BIOS execute.
                        ;
0003                    STOP            EQU     03H             ; STOP code
000A                    LF              EQU     0AH             ; Line feed
000D                    CR              EQU     0DH             ; Carriage return
                        ;       ************************************************
                        ;               MAIN PROGRAM
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;               This program is setting power off & alarm
                        ;               alarm disable, and if key inputed, do power
                        ;               off or alarm.
                        ;
0100                    MAIN:
0100    31 1000                 LD      SP,1000H        ; Set stack pointer.
                        ;
0103    CD 0121                 CALL    DISABLE         ; Interrupt disable.
                        ;
0106                    MAIN10:
0106    3A 0237                 LD      A,(PEND)        ; Stop key check.
0109    B7                      OR      A               ; Stop key pressed?
010A    C2 011B                 JP      NZ,MAIN20       ; Yes.
                        ;
010D    76                      HALT                    ; Wait until interrupt happened.
010E    CD 0144                 CALL    CHKINT          ; Check interrupt status.
0111    28 F3                   JR      Z,MAIN10        ; Neither power off nor alarm
                        ;
0113    F5                      PUSH    AF              ; Save interrupt information.
0114    CD 0185                 CALL    KEYIN           ; Message display and key in.
0117    F1                      POP     AF              ; Restore interrupt information.
                        ;
0118    CD 019F                 CALL    OKINT           ; Interrupt execute.
                        ;
011B                    MAIN20:
011B    CD 0132                 CALL    ENABLE          ; Interrupt enable.
011E    C3 EB03                 JP      WBOOT           ; End.
                        ;
                        ;       ************************************************
                        ;               DISABLE POWER OFF & ALARM
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;               Disable the following system function.
                        ;                1. Power off execute.
                        ;                2. Alarm screen display.
                        ;
                        ;       <> entry parameter <>
                        ;               NON
                        ;       <> return parameter <>
                        ;               NON
                        ;       <> preserved registers <>
                        ;               NON
                        ;
                        ;       CAUTION :
                        ;
0121                    DISABLE:
0121    3A EFEF                 LD      A,(YPOFDS)      ; Set power off disable.
0124    F6 01                   OR      01H             ; Bit 0 is application bit.
0126    32 EFEF                 LD      (YPOFDS),A      ;
                        ;
0129    3A EFF1                 LD      A,(YALMDS)      ; Set alarm disable.
012C    F6 01                   OR      01H             ; Bit 0 is application bit.
012E    32 EFF1                 LD      (YALMDS),A      ;
0131    C9                      RET                     ;
                        ;
```

```
                        ;*************************************************
                        ;       ENABLE POWER OFF & ALARM INTERRUPT
                        ;*************************************************
                        ;
                        ;       NOTE :
                        ;
                        ;       <> entry parameter <>
                        ;               NON
                        ;       <> return parameter <>
                        ;               NON
                        ;       <> preserved registers <>
                        ;               NON
                        ;
                        ;       CAUTION :
                        ;
0132                    ENABLE:
0132    21 EFEF         ;       LD      HL,YPOFDS       ; Reset my disable bit.
0135    CB 86           ;       RES     0,(HL)          ;
                        ;
0137    21 EFF1         ;       LD      HL,YALMDS       ; Reset my disable bit.
013A    CB 86           ;       RES     0,(HL)          ;
                        ;
013C    CD 0144         ;       CALL    CHKINT          ; Check interrupt happened.
013F    C8              ;       RET     Z               ; No interrupt.
0140    CD 019F         ;       CALL    OKINT           ; Interrupt execute.
                        ;
0143    C9              ;       RET                     ;
                        ;
                        ;*************************************************
                        ;       CHECK POWER OFF & ALARM INTERRUPT
                        ;*************************************************
                        ;
                        ;       NOTE :
                        ;               Check power off & alarm interrupt occurred.
                        ;               If occurred, set the information to return
                        ;               code.
                        ;
                        ;       <> entry parameter <>
                        ;               NON
                        ;       <> return parameter <>
                        ;               Z-flag  : Return information.
                        ;                       =0 -- Both interrupt not occurred.
                        ;                       =1 -- Interrupr occurred
                        ;               A       : Interrupt type
                        ;                 bit 0 : Alarm interrupt.
                        ;                 bit 1 : Power off interrupt.
                        ;                       ( 1=occurr, 0=not occurr)
                        ;
                        ;       <> preserved registers <>
                        ;               NON
                        ;
                        ;       CAUTION :
                        ;               If STOP key is pressed, then sets
                        ;               PEND flag.
                        ;
0144                    CHKINT:
0144    CD EB06         ;       CALL    CONST           ; Key in check.
0147    3C              ;       INC     A               ; No inputed key?
0148    20 0A           ;       JR      NZ,CHK10        ; Yes.
                        ;
014A    CD EB09         ;       CALL    CONIN           ; Get key code.
014D    FE 03           ;       CP      STOP            ; STOP key?
014F    20 03           ;       JR      NZ,CHK10        ; No.
                        ;
0151    32 0237         ;       LD      (PEND),A        ; Set program end flag.
                        ;
0154                    CHK10:
0154    0E 00           ;       LD      C,00H           ; Clear return information.
                        ;
0156    3A EFF0         ;       LD      A,(YPOFST)      ; Check power off status.
0159    B7              ;       OR      A               ; Power off occurred?
015A    28 10           ;       JR      Z,CHK20         ; No.
                        ;
015C    21 EFEF         ;       LD      HL,YPOFDS       ; Reset my disable bit.
015F    CB 86           ;       RES     0,(HL)          ;
                        ;
0161    E6 FE           ;       AND     11111110B       ; Reset my status bit.
0163    32 EFF0         ;       LD      (YPOFST),A      ;
0166    20 04           ;       JR      NZ,CHK20        ; Disable by other.
                        ;
0168    79              ;       LD      A,C             ; Set Power-off- go bit.
0169    F6 02           ;       OR      02H             ;
016B    4F              ;       LD      C,A             ;
                        ;
016C                    CHK20:
016C    3A EFF2         ;       LD      A,(YALMST)      ; Check alarm status.
016F    B7              ;       OR      A               ; Alarm occurred?
0170    28 10           ;       JR      Z,CHK40         ; No.
                        ;
0172    21 EFF1         ;       LD      HL,YALMDS       ; Reset my disable bit.
0175    CB 86           ;       RES     0,(HL)          ;
                        ;
0177    E6 FE           ;       AND     11111110B       ; Reset my status bit.
0179    32 EFF2         ;       LD      (YALMST),A      ;
017C    20 04           ;       JR      NZ,CHK40        ; Disable by other.
                        ;
017E    79              ;       LD      A,C             ; Set alarm-go bit.
017F    F6 01           ;       OR      01H             ;
0181    4F              ;       LD      C,A             ;
                        ;
0182                    CHK40:
0182    79              ;       LD      A,C             ; Set return information.
0183    B7              ;       OR      A               ;
0184    C9              ;       RET                     ;
                        ;
                        ;
```

```
                        ;       ************************************************
                        ;               STATUS DISPLAY & KEY IN
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;               If power off or alarm occurred, then
                        ;               display message & wait until key inputed.
                        ;
                        ;       <> entry parameter <>
                        ;               A  : Interrupt type.
                        ;                  bit 0 : alarm
                        ;                  bit 1 : power off
                        ;       <> return parameter <>
                        ;               NON
                        ;       <> preserved registers <>
                        ;               NON
                        ;
                        ;       CAUTION :
                        ;
0185                    KEYIN:
0185    0F              ;       RRCA                    ; Alarm bit --> CY
0186    F5              ;       PUSH    AF              ; Save interrupt status.
0187    21 01D3         ;       LD      HL,MSG01        ; Alarm happen message.
018A    DC 01C7         ;       CALL    C,DSPMSG        ; Display if alarm occurred.
018D    F1              ;       POP     AF              ; Restore interrupt status.
018E    0F              ;       RRCA                    ; Power off bit --> CY
018F    21 01EF         ;       LD      HL,MSG02        ; Power off happen message.
0192    DC 01C7         ;       CALL    C,DSPMSG        ; Display if power off occurred.
                        ;
0195    21 021A         ;       LD      HL,MSG03        ; Key in message
0198    CD 01C7         ;       CALL    DSPMSG          ;
                        ;
019B    CD EB09         ;       CALL    CONIN           ; Input any key.
019E    C9              ;       RET                     ;
                        ;
                        ;       ************************************************
                        ;               POWER OFF OR ALARM EXECUTE
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;               Power off or alarm execute in
                        ;               this routine.
                        ;
                        ;       <> entry parameter <>
                        ;               A   : Interrupt type.
                        ;                   bit 0 : alarm
                        ;                   bit 1 : power off
                        ;       <> return parameter <>
                        ;               NON
                        ;       <> preserved registers <>
                        ;               NON
                        ;
                        ;       CAUTION :
                        ;
019F                    OKINT:
019F    F5              ;       PUSH    AF              ; Save interrupt information.
01A0    CB 4F           ;       BIT     1,A             ; Power off?
01A2    28 13           ;       JR      Z,OK20          ; No.
                        ;
01A4    0E 00           ;       LD      C,00H           ; Set continue power off mode.
01A6    3A EFEE         ;       LD      A,(BTRYFG)      ; Power fail check.
01A9    B7              ;       OR      A               ; Power fail?
01AA    20 08           ;       JR      NZ,OK10         ; Yes.
                        ;
01AC    3A EFF0         ;       LD      A,(PWSWOFFG)    ; Power off check.
01AF    FE 02           ;       CP      02H             ; Continue power off?
01B1    28 01           ;       JR      Z,OK10          ; Yes.
                        ;
01B3    0C              ;       INC     C               ; Set restart power off.
                        ;
01B4                    OK10:
01B4    CD EB7E         ;       CALL    POWEROFF        ; Go power off.
                        ;
01B7                    OK20:
01B7    F1              ;       POP     AF              ; Restore interrupt information.
01B8    CB 47           ;       BIT     0,A             ; Alarm?
01BA    C8              ;       RET     Z               ; No.
                        ;
01BB    3A EFF2         ;       LD      A,(YALMST)      ; Set BIOS bit
01BE    F6 80           ;       OR      10000000B       ;
01C0    32 EFF2         ;       LD      (YALMST),A      ;
01C3    CD FF96         ;       CALL    RSPSTBIOS       ; Go alarm.
                        ;
01C6    C9              ;       RET                     ;
                        ;
                        ;       ************************************************
                        ;               DISPLAY MESSAGE UNTIL FIND 00H
                        ;       ************************************************
                        ;
                        ;       NOTE :
                        ;
                        ;       <> entry parameter <>
                        ;               HL : Message top address.
                        ;       <> return parameter <>
                        ;               NON
                        ;       <> preserved registers <>
                        ;               NON
                        ;
                        ;       CAUTION :
                        ;
01C7                    DSPMSG:
01C7    4E              ;       LD      C,(HL)          ; Get message data.
01C8    0C              ;       INC     C               ;
01C9    0D              ;       DEC     C               ; End of message?
01CA    C8              ;       RET     Z               ; Yes.
                        ;
01CB    E5              ;       PUSH    HL              ; Save pointer.
```

```
01CC    CD EB0C                 CALL    CONOUT          ; Message display.
01CF    E1                      POP     HL              ; Restore pointer.
01D0    23                      INC     HL              ; Poiner update.
01D1    18 F4                   JR      DSPMSG          ; Loop
                        ;
                        ;       Message and work area
                        ;
01D3                    MSG01:
01D3    41 6C 61 72             DB      'Alarm interrupt occurred.'
01D7    6D 20 69 6E
01DB    74 65 72 72
01DF    75 70 74 20
01E3    6F 63 63 75
01E7    72 72 65 64
01EB    2E
01EC    0D 0A 00                DB      CR,LF,00H
                        ;
01EF                    MSG02:
01EF    50 6F 77 65             DB      'Power switch off or Power fail occurred.'
01F3    72 20 73 77
01F7    69 74 63 68
01FB    20 6F 66 66
01FF    20 6F 72 20
0203    50 6F 77 65
0207    72 20 66 61
020B    69 6C 20 6F
020F    63 63 75 72
0213    72 65 64 2E
0217    0D 0A 00                DB      CR,LF,00H
                        ;
021A                    MSG03:
021A    50 72 65 73             DB      'Press any key to continue.'
021E    73 20 61 6E
0222    79 20 6B 65
0226    79 20 74 6F
022A    20 63 6F 6E
022E    74 69 6E 75
0232    65 2E
0234    0D 0A 00                DB      CR,LF,00H
                        ;
                        ;
0237                    PEND:
0237    00                      DB      00H             ; Program end flag.
                        ;
                                END
```

4.7.3.6 Extending interrupt processing

Interrupt processing can be extended by:
1. Using interrupt hooks.
2. Rewriting the interrupt vector.
3. Checking the interrupt state.

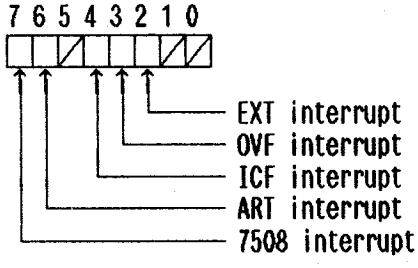
(1) Extending interrupt processing using interrupt hooks.
The interrupt hooks that the PINE supports include ICFHOOK, OVFHOOK, EXTHOOK, and HK8251 (OS kana V2.Ø (in Japan) only). See Section 4.3, "Hooks" for the location and the user of these hooks.

(2) Extending interrupt processing by rewriting the interrupt vector
User-supplied interrupt processing can be executed by rewriting the interrupt jump vector (ØFFFØH through ØFFFFH). The user must exercise care with the following when rewriting the interrupt jump vector:
1. Rewrite the jump vector after disabling the CPU for interrupts (DI state).
2. Place the new interrupt processing routine in the resident area between ØEØØØH through ØFFFFH, that is, the new vector entry must point to somewhere between ØEØØØH and ØFFFFH.
3. The interrupt processing routine must reserve its own stack area.
4. The interrupt processing routine must not use BDOS or BIOS.
5. On exit, restore the registers into the original state established when it is entered (in other words, all registers must be preserved).

(3) Checking the interrupt state
With this method, the interrupt processing routine does not perform interrupt processing immediately but checks for interrupts frequently. The occurrence of interrupts can be tested by examining the following:
1. Interrupt flag INTTYPE (ØEFD3H)
2. Interrupt Status Register (ISR) (PØ4H)

See Part I, "Firmware" for ISR.

INTTYPE (ØEFD3H) 1 byte
Interrupt flags



When an interrupt occurs, the corresponding bit position is set to 1.

4.7.4 7508 Interrupts

4.7.4.1 Outline

When an interrupt is generated by the 7508, the interrupt processing routine reads the 7508 status through the serial communication interface, and transfers control to the corresponding routine that services the interrupt. The routines that service 7508 interrupts include:

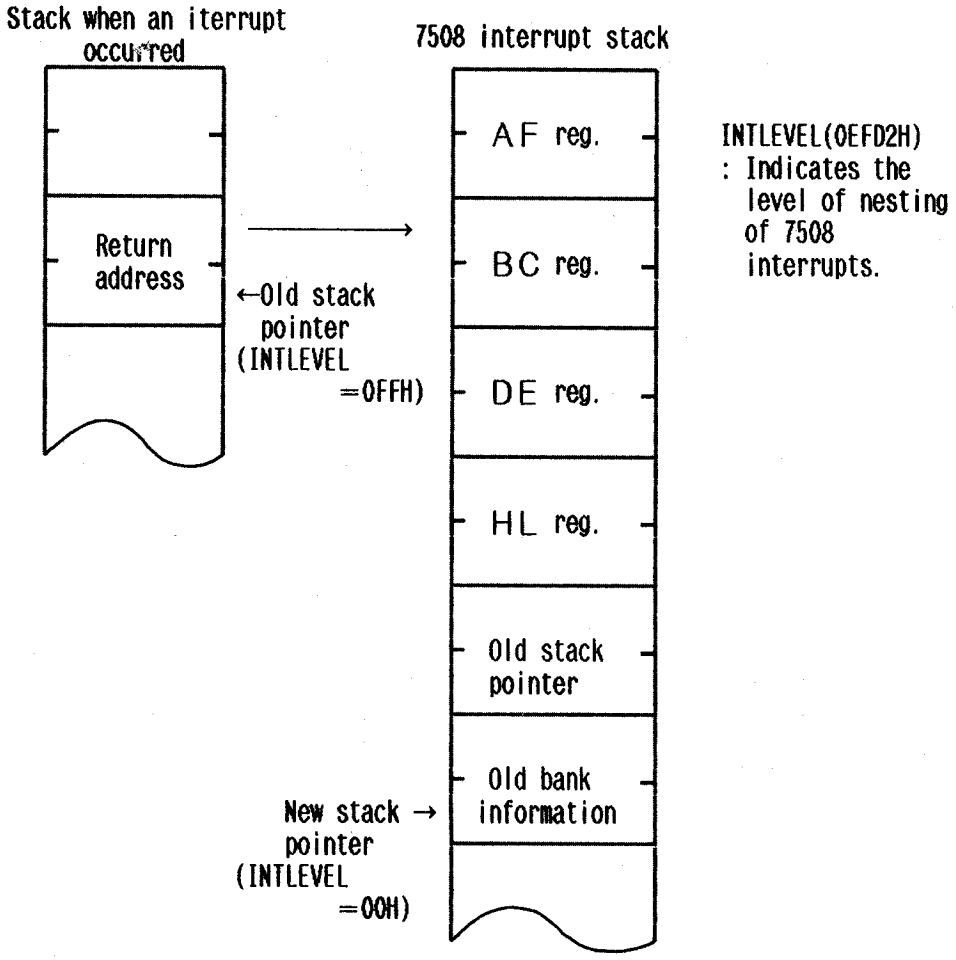
- Key interrupt
- 1-second interrupt
- Power fail interrupt
- Alarm interrupt
- Power switch interrupt

7508 interrupt processing allows multiple 7508 interrupts so that the PINE can accept key inputs while displaying the alarm screen.

4.7.4.2 Multiple interrupt processing

Provisions are made in 7508 interrupt processing so that multiple 7508 interrupts can be honored.

Multiple interrupt processing is accomplished using the interrupt level work area INTLEVEL (0EFD2H) and stacks. The stacks used for multiple interrupt processing are illustrated below.



The stack status in the above figure shows the one when a 7508 interrupt occurred. If another 7508 interrupt occurs in this condition, the interrupt processing routine does not switch the stacks, but saves the contents of 7508 interrupt stack, from AF register to old bank information, and increments INTLEVEL by 1.

Fig. 4.7.9 Multiple 7508 Interrupt Processing

4.7.4.3 Types of 7508 interrupts

There are five types of 7508 interrupts which can be identified by reading the 7508 status.

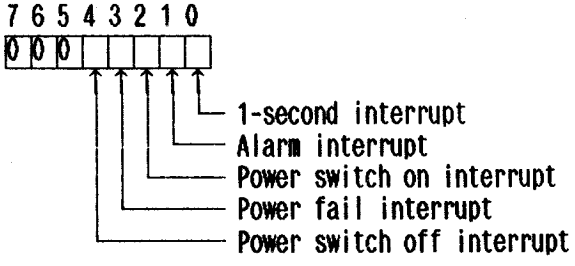
When an interrupt is generated by the 7508, the system issues a Status Read command (02H) to the 7508 to read the 7508 status. A status value 0BFH or smaller indicates a hard code generated by a key interrupt and a status value 0C0H or greater indicates a power switch, 1-second, alarm, or power fail interrupt. The system stores this status in STS7508 (0F4D6H).

Power switch interrupts are further divided into power switch on and off interrupts. After loading the status of the interrupt (power switch on/off, 1-second, alarm, or power fail interrupt), the system transfers control to the corresponding interrupt servicing routine. The interrupt servicing routine to be started is determined from the status information in INTFG (0F4D5H) and the table TBL7508 at 0EFDEH.

The status of 1-second and alarm interrupt processing is stored in FG7508 at 0F4D7H.

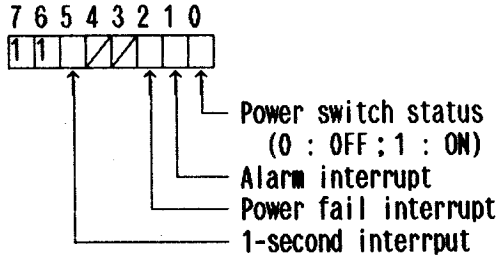
The work areas related to 7508 interrupt processing are given below.

INTFG (0F4D5H) 1 byte
Indicates the type of 7508 interrupt processing performed by the system. INTFG is set to 00H when key interrupt processing is performed.



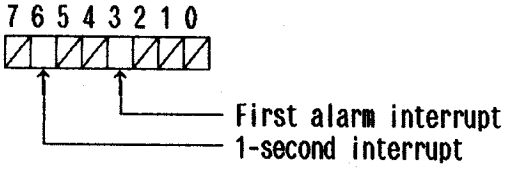
A 1 in a bit position indicates that the corresponding interrupt occurred.

STS7508 (0F4D6H) 1 byte
Contains the status read by the system from the 7508 when a 7508 interrupt occurred. Values greater than or equal to 00H and smaller than or equal to 0BFH identify key interrupts and represent keys' hard codes.
When STS7508 contains a value greater than or equal to 0C0H:



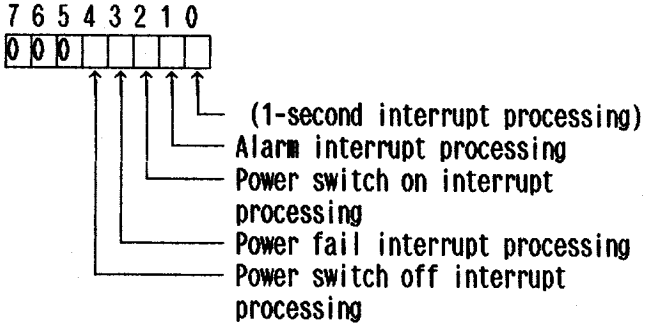
A 1 in a bit position indicates that the corresponding interrupt occurred.

FG7508 (ØF4D7H) 1 byte
Is the 7508 interrupt processing flag.



A 1 in a bit position indicates that the corresponding interrupt occurred.

TBL7508 (ØEFDEH) 16 bytes
Is the table used to determine the 7508 interrupt processing to be serviced. Each byte has the following format:



A 1 in a bit position indicates that the corresponding interrupt occurred.

The meaning of the table entries are listed below.

Address	7508 status	PW switch status	initial value
EFDEH	C0H or E0H	OFF	00H
DF		ON	10H
E0	C1H or E1H	OFF	04H
E1		ON	00H
E2	C2H or E2H	OFF	02H
E3		ON	12H
E4	C3H or E3H	OFF	04H
E5		ON	02H
E6	C4H or E4H	OFF	08H
E7		ON	10H
E8	C5H or E5H	OFF	08H
E9		ON	08H
EA	C6H or E6H	OFF	08H
EB		ON	10H
EC	C7H or E7H	OFF	08H
ED		ON	08H

The presence or absence of the 1-second interrupt processing routine can be determined by software after rewriting the table.

4.7.4.4 Key interrupt processing

When an interrupt is generated by the 7508, the system issues a Status Read command (02H) to the 7508 to read the 7508 status. A status value 0BFH or smaller indicates that a key interrupt occurred and it is processed as such.

When a key interrupt is generated by the 7508, the system interprets the status data as a hard code which indicates the position of the key on the keyboard matrix. The key interrupt processing routine places the key's hard code into the key buffer (KBUF). If the key buffer is full, the key code is discarded.

If the hard code received from the 7508 is the one for the STOP key (10H for the standard keyboard and 0B6H for the item keyboard), the system performs special processing different from that for ordinary keys.

See Section 3.5, "Keyboard" for the key buffer and hard codes.

(1) STOP key special processing
Interrupt processing proceeds as follows when the system receives a hard code for the STOP key from the 7508:

a) Sends a Keyboard Clear command to the 7508 to clear the 7-byte buffer of the 7508, and waits for approximately 40 ms (key scan time). If an item keyboard is installed or the STOP and CTRL/STOP keys are disabled (specified in ISYSREG (0F01EH)), the system does no special processing but carries out normal key interrupt processing.
b) Reads the 7508 buffer to check whether the current key is pressed with the SHIFT key.
c) Loads 0FFH into the BRKFLG (0F019H) flag that indicates that the STOP key has been pressed.

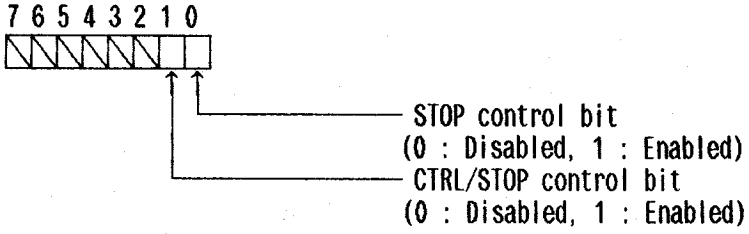
d) If the CTRL key is also pressed, the system loads ØlH into CSTOPFLG (ØFØlAH), which is a flag indicating that the CTRL and STOP keys have been pressed simultaneously, CSTOPMCT (ØFØlBH), and CSTOPPRN (ØF3Ø9H). The system bypasses this step if an item keyboard is installed or the CTRL and STOP keys are disabled.

e) Checks the state of the switch key that is currently being pressed to determine the keyboard state.

f) Clears the key buffer (KBUF) and loads the hard code (lØH) for the STOP key into the key buffer. The system places lØH into the key buffer also when an item keyboard is installed.

The work areas related to key interrupt processing are described below.

ISYSREG (ØFØlEH) l byte
Is the system function flag.



BRKFLG (ØF9lH) l byte
Indicates whether the STOP key has been pressed.
= ØØH: The STOP key has not been pressed.
= ØFFH: The STOP key has been pressed.

CSTOPFLG (ØFØlAH) l byte
Indicates whether the CTRL/STOP keys have been pressed.
= ØØH: The CTRL/STOP keys have not been pressed.
= ØlH: The CTRL/STOP keys have been pressed.

CSTOPMCT (ØFØlBH) l byte
Flag for terminating microcassette processing.
= ØØH: The CTRL/STOP keys have not been pressed.
= ØlH: The CTRL/STOP keys have been pressed.

CSTOPPRN (ØF3Ø9H) l byte
Flag for terminating cartridge printer.
= ØØH: The CTRL/STOP keys have not been pressed.
= ØlH: The CTRL/STOP keys have been pressed.

Note: BRKFLG and CSTOPFLG are reset to ØØH when BIOS CONIN or CONST is called and CSTOPMCT and CSTOPPRN are reset by PSTBIOS.

4.7.4.5 1-second interrupt processing

1-second interrupt processing proceeds as follows:

a) Increments 16-bit timer TIMER0 (0EF8FH) by 1.
b) Decrements 16-bit timer TIMER1 (0EF91H) by 1.
c) Processes timer function TMFUNC (0F313H).
   When TMFUNC (0F313H) contains a nonzero value, the interrupt processing routine decrements TMSEC (0F4DAH) by 1. If the TMSEC reaches 00H as the result of the decrement, the interrupt processing routine sets TMFUNC to 00H and TMGLAG (0F314H) to 0FFH.
d) Counts down the ROM cartridge power-off time.
e) Counts down the alarm repeat time.

The areas referred to in steps a), b), and c) are 1-second counters available for application programs and those referred to in steps d) and e) are counters available for system control. TIMER0 is also used by the system to monitor the auto power-off time.

(1) Uses of TIMER0 and TIMER1

TIMER0 and TIMER1 are 16-bit 1-second counters and incremented or decremented by 1 every time a 1-second interrupt occurs. These counters can be used as reference-only counters (not overwritten) to measure processing time.

(2) Use of TMFUNC

TMFUNC is a flag area available for the user to specify the execution of the timer function for checking for the lapse of a specified time. Figure 4.7.10 shows how to use this flag.

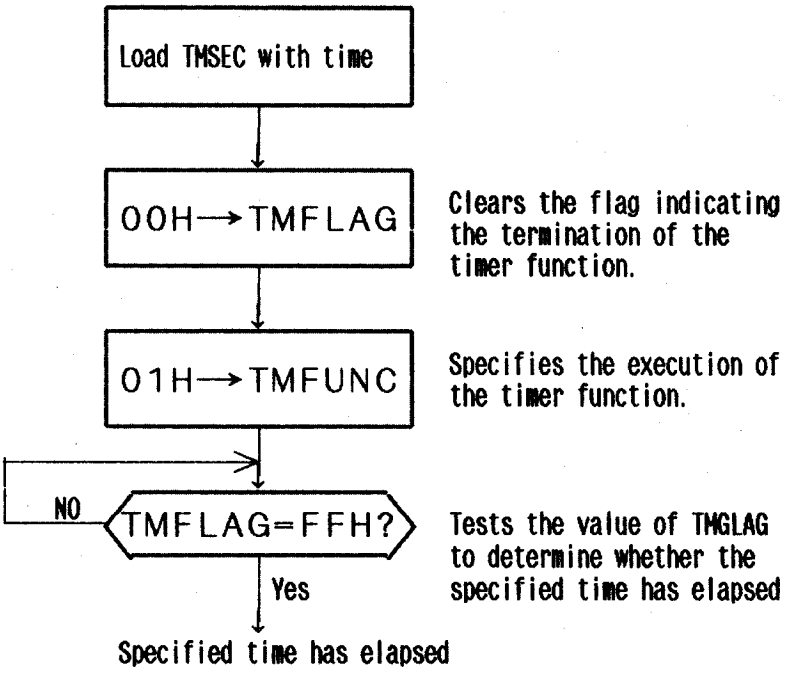


Fig. 4.7.10 Using TMFUNC

The work areas related to 1-second interrupt processing are described below.

TIMERØ (ØEF8FH) 2 bytes
16-bit timer incremented by 1 every time a 1-second interrupt occurs.

TIMER1 (ØEF91H) 2 bytes
16-bit timer decremented by 1 every time a 1-second interrupt occurs.

TMFUNC (ØF313H) 1 byte
Timer function flag.
= ØØH: The timer function is not specified.
= Nonzero: The timer function is specified.

TMFLAG (ØF314H) 1 byte
Timer flag.
When TMFUNC contains a nonzero value, TMSEC is decremented by 1 every 1-second interrupt and TMFLAG is set to ØFFH when TMSEC reaches ØØØØH.

TMSEC (ØF4DAH) 2 bytes
Timer counter.
When TMFUNC contains a zero value, TMSEC is decremented by 1 every 1-second interrupt and when TMSEC reaches ØØØØH, TMFLAG is set to ØFFH and TMFUNC to ØØH.

4.7.4.6 Alarm interrupt processing

An alarm interrupt occurs when the time specified by BIOS TIMDAT with a Set Alarm/Wake is reached. Alarm interrupts are distinguished from wake interrupts by software; on hardware, only alarm interrupts can occur.

Although the alarm time can be specified to the precision of month, day, day of the week, hour, minute, and second, the minimum unit is 1Ø seconds. This is because the 75Ø8 recognizes the arrival of an alarm time by a coincidence between the counter value and the specified time in the precision up to 1Ø seconds. Accordingly, once the set alarm time is reached, a maximum of 1Ø alarm interrupts can occur. PINE OS takes one of the 1Ø interrupts as a true alarm interrupt.

PINE OS places hooks in the alarm interrupt processing routine so that the application programs can extend the capability of the alarm interrupt processing routine. See Section 4.3, "Hooks" for details on alarm hooks.

The display of the alarm screen can be disabled by software during alarm processing using ALRMDS (ØEFF1H).

4.7.4.7 Power switch on interrupt processing

A power switch on interrupt occurs when the power switch is switched from the OFF to ON position. The power switch on processing routine only sets the power switch on flag PWSWONFG (ØF4DCH). See Section 2.4, "Power-on" for power-on processing started when the main power is off.

The work area related to power switch on interrupt processing is described below.

PWSWONFG (ØF4DCH) 1 byte
Power switch on flag.
= ØØH: No power switch on interrupt has occurred.
= ØFFH: A power switch on interrupt has occurred.
Reset to ØØH when power switch on processing is started with the main power switch in the OFF position.

4.7.4.8 Power fail interrupt processing

A power fail interrupt occurs when the main battery voltage falls below a certain level. The threshold level is 4.8 volts for NiCd batteries and 4.Ø volts for Mn dry batteries.

When a power fail interrupt occurs, the corresponding interrupt processing routine sets the power fail flag BTRYFG (ØEFEEH) to alert that the battery voltage has been fallen. See Section 2.6, "Power Fail" for details on the power fail screen.

Once a power fail interrupt occurs, the same interrupt occurs every one second. If main power is not shut down within 5Ø seconds after the first power fail interrupt, the 75Ø8 forces main power to be shut down.

The work area related to power fail interrupt processing is described below.

BTRYFG (ØEFEEH) 1 byte
Power fail interrupt flag.
= ØØH: No power fail interrupt has occurred.
= ØFFH: A power fail interrupt has occurred.
Reset to ØØH by a power-on start sequence.

4.7.4.9 Power switch off interrupt processing

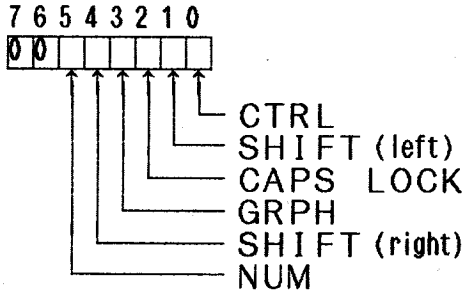
A power switch off interrupt occurs when the power switch is switched from the ON to OFF position. The power switch off processing routine tests the current key shift state to determine whether the CTRL key has been pressed, places the result into power switch off flag PWSWOFFG (ØF4DDH), then turns off power.

The key shift state that determines the continue or restart mode power-off can be altered. A continue mode power-off occurs when the value of CNTNKEY (ØEF2AH) match the key shift state when the power switch off interrupt occurred. See Section 2.5, "Power-off" for details on power off processing.

The work areas related to power switch off interrupt processing are described below.

PWSWOFFG (ØF4DDH) 1 byte
Power switch off flag.
= ØØH: No power switch off interrupt has occurred.
= Ø1H: A restart mode power switch off interrupt occurred.
= Ø2H: A continue mode power switch off interrupt has occurred.

CNTNKEY (ØEF2AH) 1 byte
Continue key shift state flag.



A 1 in a bit position indicates that the corresponding shift key is pressed.
Only bit Ø (CTRL) is valid when an item keyboard is installed.
Only continue mode power switch off interrupts are enabled when CNTNKEY contains ØØH, irrespective of the key shift state.
This area is initialized to Ø1H.

4.7.5 ART Interrupts

An ART interrupt occurs when the RxRDY pin in the ART block is set. ART interrupt processing proceeds as follows (see also Section 3.4, "BIOS Details" since ART interrupt processing is closely related to BIOS RSIOX processing):

a) Does nothing if the RSIOX OPEN function has not been performed.
b) Checks for errors (framing, receive overrun, parity, and receive buffer overflow errors).
c) Reads the received data into the receive buffer.
d) If XON/XOFF control is specified, checks the number of the received bytes in the receive buffer and sends an XOFF code if it exceeds the 3/4 of the buffer size.

4.7.6 OVF Interrupts

An OVF interrupt occurs when an FRC (Free Running Counter) overflow occurs. The FRC is a 16-bit counter running on the 616.6 kHz clock, so an OVF interrupt occurs every 1Ø6.7 msec or so.

PINE OS uses OVF interrupts for controlling cursor blinking. THe cursor turns on and off every 5ØØ msec or so. The blinking interval can be changed by rewriting BLNKTIME (ØEFBAH).

BLNKTIME (ØEFBAH) 1 byte
Specifies the cursor blink time in 1ØØ ms units. This area is initialized to Ø4H.

4.7.7 ICF/EXT Interrupts

ICF interrupts occur as the state of the input to the barcode reader or external cassette drive changes whereas EXT interrupts are generated by the interrupt signal from external devices (sent via the system bus).

When an ICF or EXT interrupt occurs, the PINE does nothing but sets interrupt flag INTTYPE at ØEFD3H. Hooks are provided for extension of ICF/EXT interrupt processing. See Section 4.3, "Hooks" for details.