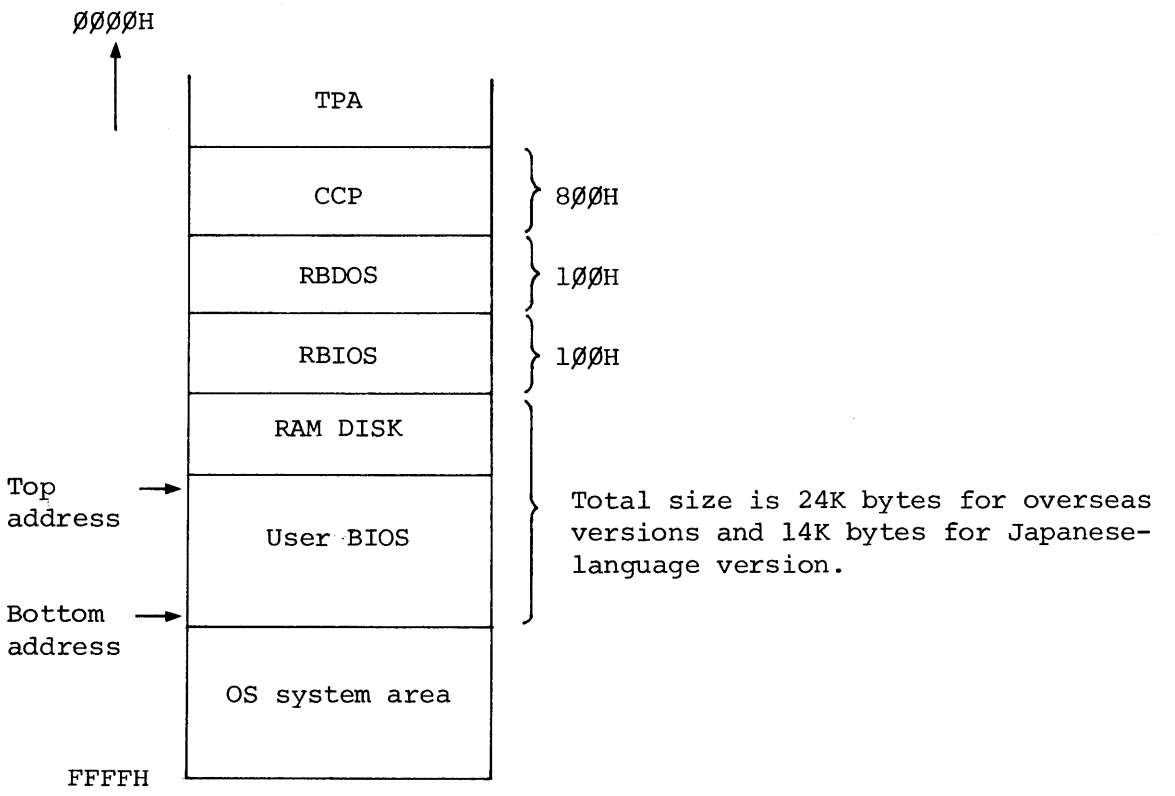
Chapter 17 How to Use User BIOS Area

17.1 Outline

A user BIOS area is reserved in MAPLE main memory to hold machine-language routines (e.g., bar code read program) or data (the scheduler stores its data here) to be shared by two or more programs.



Either the BIOS or interrupt processing routines can be extended by loading their extended portions in the user BIOS area.

17.2 User BIOS Area Specifications

The specifications of the user BIOS area are described in this section.

(1) The bottom address is determined by the system.

Overseas version (PX-8) ---------- 0EBFFH

Japanese-language version (HC-80) ---- 0E7FFH

Japanese-language version (HC-88) ---- 0ClFFH

(2) The size of the user BIOS area may be specified in 256-byte increments provided that the total size of this area and the RAM disk do not exceed the limit shown below.

Overseas version (PX-8) ---------- 24K bytes

Japanese-language version (HC-80) ---- 14K bytes

Japanese-language version (HC-88) ---- 14K bytes

(3) The user BIOS can be specified during system initialization or by using the CONFIG program.

(4) The contents of the user BIOS area are preserved until a system initialization is performed.

(5) Generally, only one program or a block of data can occupy the user BIOS area at a time. A header must be placed in the user BIOS area to have the area shared by more than one program or block of data. Details on the header are given in Section 17.3.

(6) The sizes of the user BIOS and RAM disk are loaded in the following work areas:

- USERBIOS — Loaded with the user BIOS area size specified in 256-byte units.
  - Overseas = ØF ØØBH
  - Japanese-language = ØED ØBH
- SIZRAM — Loaded with the RAM disk size specified in 1K-byte units.
  - Overseas = ØF ØØ9H
  - Japanese-language = ØED Ø9H

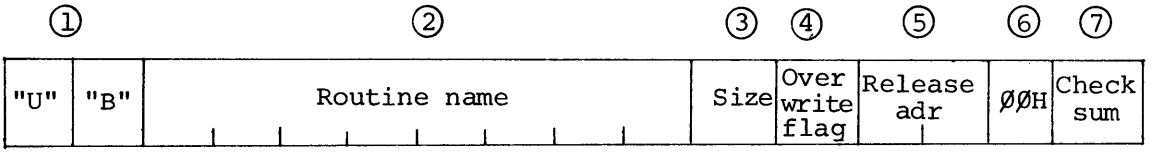
(USERBIOS) + (SIZRAM) <= 24K bytes or 14K bytes

17.3 Programming Notes on the Use of the User BIOS Area

(1) Outline

As described in the previous section, the user BIOS area may be shared by more than one program or block of data by placing a 16-byte header at the end of the area. The header is used by the application program to check whether the program or data to be used is available in the user BIOS area.

(2) Header format



The header is always located at the following address since the bottom address of the User BIOS area is fixed (the top address differs depending on the size of the user BIOS area).

Machine	OS	Address
MAPLE	ASCII	EBFØH - EBFFH
	Kana	E7FØH - E7FFH
	Japanese-language, TXT	ClFØH - ClFFH

(2.1) Header contents

No.	Item	Size (byte)	Description
1	Header ID	2	ID for the header. Fixed to "UB".
2	Routine name	8	Name of the routine loaded in the user BIOS area. Any name may be specified in ASCII. (Specify a name which is not used for another routine.)
3	Size	1	Indicates the size of the routine loaded in the user BIOS area in 256-byte units stored in binary.
4	Overwrite flag	1	Flag for indicating whether the currently loaded routine can be overwritten. ØØH: Overwrite disabled. Others: Overwrite after release processing enabled. (See the next subsection for details.)

No.	Item	Size (byte)	Description
5	Release adr.	2	The processing routine at this address is executed before a routine currently loaded in the user BIOS area is overwritten by a new routine. This release processing routine may be executed only when the overwrite flag for the currently loaded routine is set to ØØ. The release address must fall within in the user BIOS area. The release processing routine must end with a RET instruction.
6	Not used.	1	Fixed to ØØH.
7	Checksum	1	Loaded with the result obtained by subtracting the contents of the 15 bytes (from the header top to the item preceding Checksum) from ØØH, sequentially one byte at a time. (This result is used for checking the validity of the header data.)

(3) Overwrite flag

1) ØØH: For routines which inhibit loading of new routines

Set the overwrite flag to ØØH when loading a routine which must be resident in the user BIOS area (scheduler resident routines, for example) once it is loaded.

This routine can be deleted from the user BIOS area only by the program that loaded the routine.

2) Other values: For routines which allow loading of new routines after execution of a release processing routine

A nonzero value must be specified to allow a new routine to be loaded into the user BIOS area when the user BIOS area can be restored into the original state after the execution of a release processing routine.

Set this flag to a nonzero value for user routines (e.g., routines which change the contents of a hook or

jump vector table) which alter the system area at load time but which can restore the system area into the original state by executing the release processing routine and load a new routine into the user BIOS area.

(4) Release processing routine

A user BIOS routine which is to modify the contents of the system area (hook or jump table, for example) must save the original contents of the system area into the user BIOS area before starts execution.

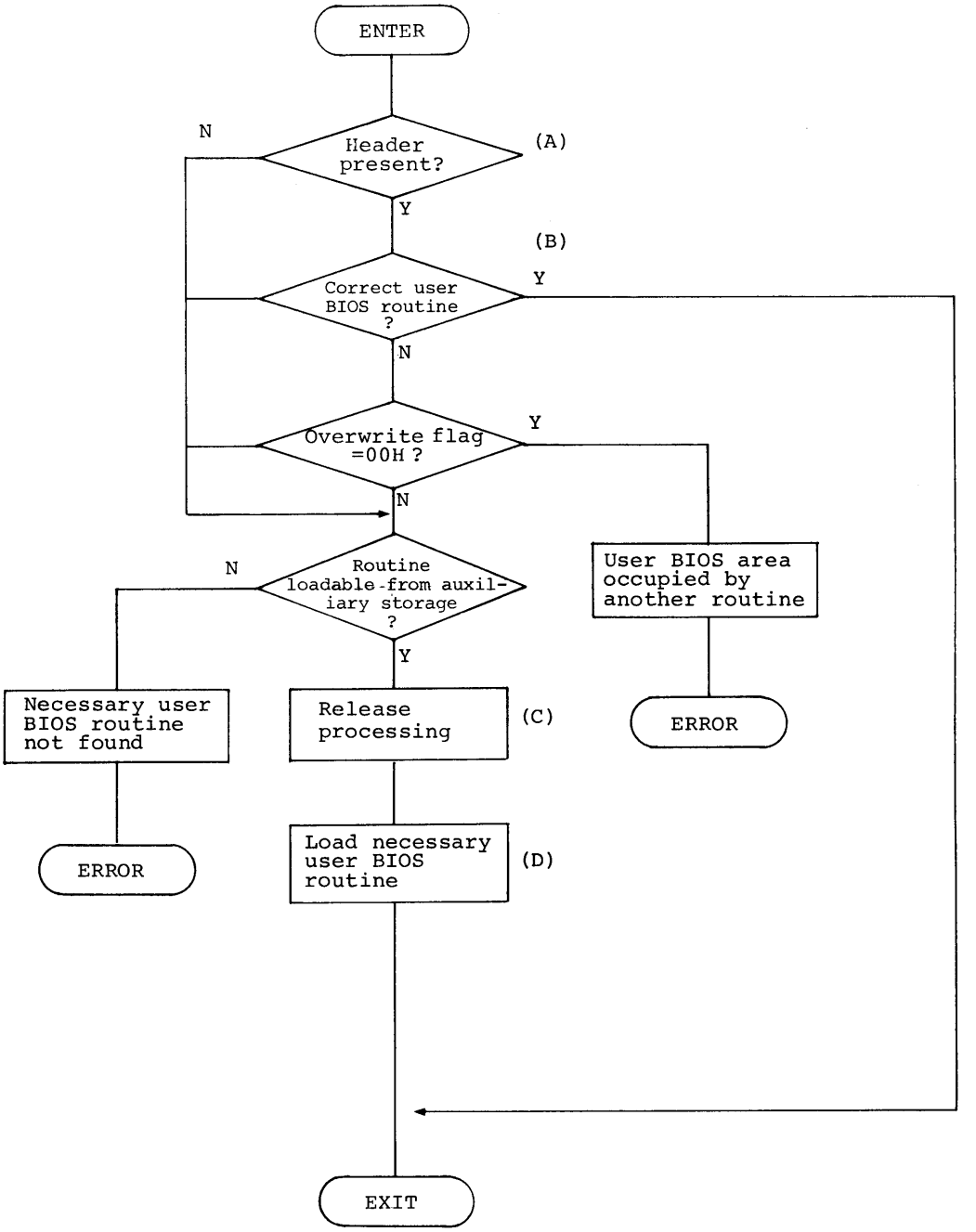
The release processing routine is called to restore the system into the state before the user BIOS routine is loaded by placing the saved contents back into the system area and setting all header fields to ØØHs.

The header must be cleared even if the system area need not be restored to the original state.

Note: Place the release processing routine in the highest 256 bytes (including the header) of the user BIOS area.

(5) Procedure for using the routine in the user BIOS area in the application

The application program must verify that the user BIOS routine is available before accessing that routine. The procedure shown below must be observed to check this.



(A) Check whether the correct header is present by matching the header ID with "UB" and checksum. If the header ID field contains "US", it is unconditionally concluded that the scheduler is using the user BIOS area because the MicroPro scheduler for MAPLE defines "US" as the header ID.

(B) Check to determine whether the required user BIOS routine is loaded in the user BIOS area by checking the routine name in the header.

(C) Call the routine addressed by the release address in the header.

(D) Load a new routine to the user BIOS area and update the header contents.