

Chapter 4

6502 BIOS

Installing User-Written
Software in the 6502 BIOS 94

6502 BIOS Operation 94

Changing the 6502 BIOS 96

6502 Memory Map 97

Implementing Your Own Software 98

6502 BIOS Call Descriptions 99

0 CALLSUB 100

1 WRITEMEM 101

2 READMEM 102

3 READSEC 103

4 WRITESEC 104

5 READSLOT 105

6 WRITESLOT 106

7 STATSLOT 107

8 INITSLOT 108

9 WSTART 109

10	FORMAT	110
11	UPDATE	111
12	BEEP	112
13	CLEAR	113
14	INVERT	114
15	SETPT1	115
16	SETPT2	116

This chapter describes the 17 functions requests, the 6502 BIOS calls, that access the 6502 microprocessor. A listing of the system calls is provided in the following table.

Table 4.1.

6502 BIOS Calls

Call Number	Name	Call Number	Name
0	CALLSUB	9	WSTART
1	WRITEMEM	10	FORMAT
2	READMEM	11	UPDATE
3	READSEC	12	BEEP
4	WRITESEC	13	CLEAR
5	READSLOT	14	INVERT
6	WRITESLOT	15	SETPT1
7	STATSLOT	16	SETPT2
8	INITSLOT		

The guidelines for using the 6502 BIOS calls are the same as the guidelines for using CP/M system calls, except that parameters are transferred in and out of a seven-byte block of memory (SoftCard addresses 0045H—004BH) instead of the CPU registers. See “6502 BIOS Calls” in Chapter 2 for details.

Installing User-Written Software in the 6502 BIOS

You can install your own device drivers or other user-written software as part of the 6502 BIOS with the Premium SoftCard IIe. This section describes the facilities and programming conventions you will need to perform this task. Strict adherence to the programming conventions will ensure compatibility between user-written programs in the 6502 BIOS.

Important

Before attempting to install software in the 6502 BIOS, you should have experience in assembly language programming and be familiar with both Z80 and 6502 instruction sets.

6502 BIOS Operation

When a SoftCard IIe is installed in the Apple IIe and CP/M is loaded into memory, both the Z80 and the 6502 microprocessors are run simultaneously. The Z80, however, has executive control over the system. Because the Z80 cannot address the 6502 RAM directly and the Apple IIe uses memory-mapped I/O, the Z80 must use the 6502 to perform all I/O operations. The 6502 software that performs the I/O processing is called the 6502 BIOS.

The SoftCard IIe BIOS consists of two parts: the Z80 or CP/M BIOS, which interfaces to the CP/M operating system, and the 6502 BIOS which controls the I/O devices and implements the print spooler.

Usually the 6502 BIOS, represented by Figure 4.1, is in a loop waiting for a command from the Z80. When the 6502 receives a command, it sends a command back to halt the Z80. The 6502 BIOS then performs a setup routine and jumps (CMDJMP) to the main command handling routine (DOCMD). When the main command handling routine is finished, it executes a clean-up routine (CMDONE). The CMDONE cleanup routine passes parameters to the Z80 memory and starts the Z80.

```

CMDLP:                                ;Main 6502 BIOS loop (waits
      .                               ;for a command from the Z80).
      .
CMDINI  .                               ;Get a command, perform setup
      .                               ;processing.
      .
CMDJMP: JMP DOCMD                     ;Jump to the main command
      .                               ;handling routine.
      .
      .
DOCMD:  .                               ;Execute the command.
      .
      .
      .
CMDONE: .                               ;Perform cleanup routine
      .                               ;and turn on the Z80.
      .
      .
      JMP CMDLP                       ;Jump to CMDLP and wait for
      .                               ;the next Z80 command.

```

Figure 4.1. A Simplified Representation of 6502 BIOS

Changing the 6502 BIOS

The 6502 destination address of CMDJMP and the beginning address of CMDONE are stored in Z80 memory locations CMDVEC and CMDEXT. This allows you to implement your own software in the 6502 BIOS. If you replace the destination address of CMDJMP with the 6502 address of your own program or driver, control is passed to your program instead of the main command routine when a 6502 BIOS call is made. At this point, your program must decide whether to process the call or to let the main 6502 command routine process the call. Use the following code segment to define CMDEXT and CMDVEC in Z80 memory:

```
CMDEXT: DW CMDONE      ;CMDEXT is at 0F38EH
CMDVEC: DW CXMJMP+1    ;CMDVEC is at 0F390H
```

If your program processes the call, the program should jump to CMDONE when it is finished. If it lets the 6502 BIOS perform its usual processing for this call, the program will then jump to DOCMD.

6502 Memory Map

Two blocks in Figure 4.2 show the areas in the 6502 memory that are reserved for user-written programs. The rest of the 6502 memory space is reserved for the 6502 BIOS, the print spooler, text and graphics screens, and various other Apple hardware interfaces.

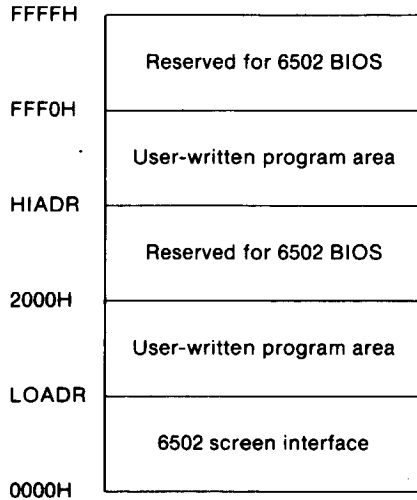


Figure 4.2. 6502 BIOS Memory Map

The highest address available in the first user area is 1FFFH. In the second user area, the highest address is FFF0H. The low addresses for each area are dependent on which routines have already been implemented in this area. Location LOMEM in Z80 memory contains the lowest address currently available in the first area, while HIMEM contains the lowest addresses available in the second area. The Z80 addresses of both locations are listed in Table 4.2.

Table 4.2.**6502 BIOS Vector Table**

Name	Z80 Address	Function
HIMEM	F394H	Contains the lowest address of the high 6502 free memory area.
LOMEM	F392H	Contains the beginning address of the low 6502 free memory area.
CMDVEC	F390H	Contains the destination address of the JMP instruction to the main 6502 BIOS call handling routine (CMDJMP).
CMDEXT	F38EH	Contains the destination address of the JMP instruction to 6502 BIOS cleanup subroutine (CMDONE).

Implementing Your Own Software

To install your own program or driver in 6502 memory, use LOMEM or HIMEM to determine the amount of available memory for your routine. Use 6502 BIOS call 1, WRITEMEM, to write your program into the designated 6502 user area, starting at the location contained in either LOMEM or HIMEM. The last step is to change the value of LOMEM or HIMEM to point to the byte following the last byte of your program or driver. If you don't update LOMEM or HIMEM, the next user routine or program that is installed could overwrite your program. Use the following code segments to define CMDEXT and CMDVEC in Z80 memory:

```
LOMEM: DW LOADR    ;LOMEM is at F392H
HIMEM: DW HIADR    ;HIMEM is at F394H
```


6502 BIOS Call Descriptions

In each of the 6502 BIOS call descriptions, a table is provided to show which parameters are needed for each call and the addresses they are stored in. Each system call includes a table of parameters showing the initial values and the returned values (if any). For example, in the following table:

Parameter	Address	Contents
Entry point	49H	2
Entry value	4AH	Low part of 6502 address
	4BH	High part of 6502 address
Returned value	45H	Data byte read from 6502 address

SoftCard memory address 49H contains the BIOS call number. (The starting point of the function request routine.) The entry value shows the type of information needed to make the call and the memory locations that are relevant to the call. (All 6 locations can be used for entry values if needed.) The returned value parameter shows the SoftCard address and data returned after each system call is made.

0 CALLSUB

Purpose

Calls a 6502 subroutine.

Parameters

Parameter	Address	Contents
Entry point	49H	0
Entry value	45H	6502 register A
	46H	6502 register X
	47H	6502 register Y
	4AH	Low part of 6502 subroutine address
	4BH	High part of 6502 subroutine address
Returned value	45H	6502 register A
	46H	6502 register X
	47H	6502 register Y
	48H	6502 status register

Remarks

A 6502 subroutine is executed with a 6502 JSR instruction. Before the JSR instruction is executed, the 6502 registers are loaded from the Z80 register pass area. At the same time, the Apple monitor ROM is banked in. When the subroutine has run, the contents of the 6502 registers are stored in the same register pass area as before (addresses 45H—4BH) and control is returned to the Z80.

1 WRITEMEM

Purpose

Writes a byte to a 6502 memory location.

Parameters

Parameter	Address	Contents
Entry point	49H	1
Entry value	45H	Data byte to be written
	4AH	Low part of 6502 address
	4BH	High part of 6502 address
Returned value	—	None

Remarks

A byte stored at SoftCard address 45H is written to the indicated 6502 address.

2 READMEM

Purpose

Reads a byte from 6502 memory.

Parameters

Parameter	Address	Contents
Entry point	49H	2
Entry value	4AH	Low part of 6502 address
	4BH	High part of 6502 address
Returned value	45H	Data byte read from 6502 address

Remarks

READMEM reads the contents of the 6502 address contained in SoftCard memory locations 4AH and 4BH. The value is returned in memory location 45H.

3 READSEC

Purpose

Reads from a disk sector.

Parameters

Parameter	Address	Contents
Entry point	49H	3H
Entry value	45H	Track number (0—34)
	46H	Disk drive number (1 or 2)
	47H	Slot number (4—6)
	48H	Sector number (0—15)
	4AH	Low part of 6502 disk sector address
	4BH	High part of 6502 disk sector address
Returned value	45H	Error return code
		0 = no error
		16 = write-protect error
		Other = I/O error

Remarks

READSEC performs a low-level disk sector read operation. The address of the 256-byte sector is stored at SoftCard memory locations 4AH and 4BH.

4 WRITESEC

Purpose

Writes to a disk sector.

Parameters

Parameter	Address	Contents
Entry point	49H	4H
Entry value	45H	Track number (0—34)
	46H	Disk drive number (0—4)
	47H	Slot number (1—7)
	48H	Sector number (0—15)
	4AH	Low part of 6502 disk sector address
	4BH	High part of 6502 disk sector address
Returned value	45H	Error return code
		0 = no error
		16 = write-protect error Other = I/O error

Remarks

A low-level sector write is performed. The 256-byte sector is read from memory locations 4A and 4B.

5 READSLOT

Purpose

Reads a character from an accessory slot.

Parameters

Parameter	Address	Contents
Entry point	49H	5H
Entry value	47H	Slot number (1—7)
Returned value	45H	Character read

Remarks

A character is read from an accessory board installed in the indicated Apple accessory slot. The board must be of type 3, 4, or 6. See Table 6.6, "Accessory Slot Addresses and Assignments," in Chapter 6, for a description of accessory board types.

6 WRITESLOT

Purpose

Writes a character to an accessory slot.

Parameters

Parameter	Address	Contents
Entry point	49H	6H
Entry value	45H 47H	Character to be written Slot number (1—7)
Returned value	—	None

Remarks

A character is written to the accessory board in the indicated slot. The board type must be 3, 4, 5, or 6. If the board type is either 3 or 5, and the slot number is 1, then the data is buffered in the 32K-byte print buffer. See Table 6.6, "Accessory Slot Addresses and Assignments," in Chapter 6, for a description of accessory board types.

7 STATSLOT

Purpose

Gets the input status of an accessory slot.

Parameters

Parameter	Address	Contents
Entry point	49H	7H
Entry value	47H	Slot number
Returned value	47H	Slot status FFH = character ready 00H = character not ready

Remarks

If a character is ready to be read from the specified accessory slot, the value of memory location 45H will be FFH. If no character is ready, the value will be 00H.

8 INITSLOT

Purpose

Initializes a slot.

Parameters

Parameter	Address	Contents
Entry point	49H	8H
Entry value	47H	Slot number
Returned value	47H	Slot status FFH = character ready 00H = character not ready

Remarks

Initializes the accessory board in the indicated slot if the board is of type 3, 4, or 6. Other board types are unaffected. See Table 6.6, "Accessory Slot Addresses and Assignments," in Chapter 6, for a description of accessory board types.

9 WSTART

Purpose

Performs a CP/M warm start.

Parameters

Parameter	Address	Contents
Entry point	49H	9H
Entry value	—	None
Returned value	—	None

Remarks

Performs a warm start by reloading the CCP module and first 256 bytes of the BDOS module into the appropriate addresses of the SoftCard memory.

10 FORMAT

Purpose

Formats a disk.

Parameters

Parameter	Address	Contents
Entry point	49H	AH
Entry value	46H	Drive number (1 or 2)
	47H	Slot number (5 or 6)
Returned value	45H	Error return code
		0 = no error
		1—15 = I/O error
		16 = write-protect error

Remarks

Formats the disk in the indicated drive for CP/M.

11 UPDATE

Purpose

Updates keyboard definition and screen function interface tables.

Parameters

Parameter	Address	Contents
Entry point	49H	BH
Entry value	—	None
Returned value	—	None

Remarks

After changes are made to the Keyboard Definition Table, or the screen function interface tables (Software Screen Function Table or Hardware Screen Function Table) CP/M uses UPDATE to make the new information active.

12 BEEP

Purpose

Creates a tone of specified pitch and duration.

Parameters

Parameter	Address	Contents
Entry point	49H	CH
Entry value	45H	Tone duration
	36H	Tone period
Returned value	—	None

Remarks

Performs the same function as the BEEP statement in GBASIC. BEEP is intended for sound effect purposes.

13 CLEAR

Purpose

Clears the screen.

Parameters

Parameter	Address	Contents
Entry point	49H	DH
Entry value	4AH	Byte written to even screen addresses
	4BH	Byte written to odd screen addresses
Returned value	—	None

Remarks

Performs the same function as the GBASIC GR 1 command. The byte at 4AH is written to all even locations on the high-resolution graphics screen. The byte at 4BH is written to all odd locations.

For more information on the GBASIC GR1 command, see the *Microsoft BASIC Interpreter Reference Manual*.

14 INVERT

Purpose

Inverts the screen in GBASIC high-resolution screen mode.

Parameters

Parameter	Address	Contents
Entry point	49H	EH
Entry value	—	None
Returned value	—	None

Remarks

All bytes on the high-resolution graphics screen are inverted.

15 SETPT1

Purpose

Sets High-Resolution Graphics Point 1.

Parameters

Parameter	Address	Contents
Entry point	49H	FH
Entry value	46H	Exclusive OR mask
	47H	AND mask
	4AH	Low part of 6502 addresses
	4BH	High part of 6502 addresses
Returned value	—	None

Remarks

The indicated screen byte is first XORed with the data at 46, the result is then ANDed with the data at 47, and finally this result is XORed onto the screen.

16 SETPT2

Purpose

Sets High-Resolution Graphics Point 2.

Parameters

Parameter	Address	Contents
Entry point	49H	10H
Entry value	45H	Byte to exclusive OR with screen byte
	4AH	Low part of 6502 addresses
	4BH	High part of 6502 addresses
Returned value	—	None

Remarks

The data at 45 is XORed onto the screen at the 6502 memory location stored at addresses 4A and 4B.