JiffyMON/64 2.0 Machine Language Monitor User's Guide
retyped & improved by slark.me/c64

TABLE OF CONTENTS

SECTION 1
INTRODUCTION

JiffyMON/64 is a unique and powerful machine language monitor for Commodore 64 and 128 (in 64 mode) computers equipped with the JiffyDOS speed-enhancement ROMs. Like other machine language monitors, JiffyMON lets you inspect and modify individual bytes in your computer's memory; assemble, disassemble, and execute 6510 assembler code programs; and also provides a number of features which assist in debugging and writing assembly language programs. JiffyMON goes beyond average machine language monitors, however, and offers a number of additional features:

Works along with BASIC

Unlike other machine-language monitors, which force you to exit BASIC in order to use them, JiffyMON works right along with the BASIC interpreter in direct mode. This enables you to work on BASIC and assembly language programs simultaneously, without the hassle of constantly having to switch back and forth between BASIC and a ML monitor. It also enables you to use all of the powerful JiffyDOS disk-related commands while the monitor is active. In addition, because JiffyMON is active along with BASIC, you can learn more about the way BASIC works in your computer at the machine-language level.

Does not steal any BASIC or normal ML memory

Because of special provisions built into JiffyDOS, JiffyMON can reside and operate completely hidden "underneath" the Kernal ROM. This keeps JiffyMON out of normal BASIC ($0800-$9FFF) and machine language ($COOO-$CFFF) memory in the C-64, leaving you as much room as possible for your BASIC and machine-language programs.

Does not use zero page or the cassette buffer

Because using zero page is so important in your own programs, JiffyMON does not alter any zero page memory locations ($OOOO-$OOFF) during its operation. This, along with leaving the cassette buffer untouched, ensures that JiffyMON does not interfere with BASIC in any way.

Disk drive monitor built in

One of the most powerful features of JiffyMON is its ability to work on the memory inside your disk drives. Almost all of the JiffyMON commands can operate on the memory inside your 1541, 1571, or compatible disk drive, enabling you to change values in memory, assemble programs in free RAM, and disassemble the DOS built into your disk drive. In addition, JiffyMON includes special disk-related commands that transfer memory between your computer and disk drive, and read or write sectors to and from disk.

ROM and I/O switching - Hidden RAM access

JiffyMON also enables you to switch out the Kernal ROM ($E000-$FFFF), BASIC ROM ($A000-$BFFF), and the I/O Area (VIC/SID/CIA1/CIA2/Color RAM $D000-$DFFF), so you can use this "hidden" RAM for you own machine language programs. All JiffyMON commands can access these hidden memory areas.

SECTION 2
GETTING STARTED WITH JIFFYMON

Loading JiffyMON

To load JiffyMON, insert the JiffyMON disk into your disk drive, and press **SHIFT-RUN/STOP**, the JiffyDOS command to load and run the first program on disk (remember, JiffyDOS must be installed in your machine and active for JiffyMON to work). After JiffyMON loads, it will automatically start up, and display the following message:

```
JIFFYMON/64 V2.0
B*
   PC  IRQ  SR AC XR YR SP
; E073 02BD 32 30 00 00 F8
```

This lets you know that JiffyMON is active, and gives the current 6510 register status. Your computer is still in BASIC, and you can work on BASIC programs, or enter any BASIC, JiffyDOS, or JiffyMON command at this time.

If you have copied JiffyMON to another disk (its not copy-protected, by the way) and the JiffyMON loader is no longer the first program on disk, you can load JiffyMON by entering:

LOAD "JBOOT",8,1
or
%JBOOT

You can then start up JiffyMON by entering:

SYS 64715 (JiffyMON/64 2.0)

The startup display shown above will appear after you enter the SYS, indicating that JiffyMON is active.

Entering Commands

JiffyMON commands are entered just like any other command. That is, simply type in the desired JiffyMON command (see SECTION 3 COMMANDS in this manual), and press **RETURN.** BASIC and JiffyDOS commands can also be entered as usual.

Exiting JiffyMON

If you wish to stop using JiffyMON, there are two commands to deactivate it:

E (exit and cold start BASIC)
X (exit and warm start BASIC)

Entering the JiffyMON **E** command will deactivate JiffyMON and do a cold start of BASIC. The screen display is cleared, and then the BASIC power-on message will appear at the top of the screen. Any BASIC program that was resident in memory is lost (a lost program can, however, be restored by the JiffyDOS **@U** command at this time).

Entering the JiffyMON **X** command will deactivate JiffyMON and do a warm start of BASIC. This will leave BASIC programs currently in memory undisturbed, and display the READY prompt.

Restarting JiffMON

To restart JiffyMON after it has been deactivated, enter:

SYS 64715 (JiffyMON/64 2.0)

This will start up JiffyMON again (assuming it has not been damaged by writing to memory under the Kernal ROM), and display the JiffyMON startup message. Note that it is safe to enter SYS 64715 even if JiffyMON is already active.

If you press RUN/STOP-RESTORE

Pressing **RUN/STOP-RESTORE** while in BASIC will partially disable JiffyMON. That is, all JiffyMON commands will still be active, but the scrolling of the **MEMORY DISPLAY**, and **DISASSEMBLE** commands will be disabled. To re-enable the scrolling, enter:

SYS 64715 (JiffyMON/64 2.0)

Printing the output produced by JiffyMON commands

As you use JiffyMON, you will probably find yourself wanting to print out a particular memory display or disassembly. There are two ways to do this:

1. Use the JiffyDOS **CTRL-P** command to print the current screen. This method has the advantage of being easy to implement, but is inconvenient for long listings, or for output to printers other than device #4.
2. Open a channel to your printer and divert screen output to that channel. To divert output to a printer with device #4, use the BASIC commands:

   OPEN 4,4 : CMD 4

If your printer has a different device number, adjust the commands accordingly. For example, use these commands if your printer is device #5:

OPEN 5,5 : CMD 5

Once you open a channel in this manner, the output produced by JiffyMON commands will be sent to your printer (note that the output produced by BASIC and JiffyOOS commands will be sent to the printer as well). When you want to switch back to screen output, enter the commands:

PRINT#4 : CLOSE 4

SECTION 3
COMMANDS

This section describes each of the JiffyMON commands and gives examples of their use. Note that JiffyMON was designed as a clone of the popular MICROMON machine-language monitor. As a result, many of the JiffyMON commands are identical to MICROMON commands in their syntax. There are, however, a number of options and convenience features offered by JiffyMON that are absent in MICROMON. To take advantage of these features, please review the command descriptions carefully.

Each JiffyMON command is listed by its name (in alphabetical order), and includes an outline of its syntax, its available modes (many commands will work with disk drives, as well as with your computer), a description, and examples. In the syntax outline, note that required parameters are enclosed in carets, with optional parameters enclosed in brackets:

```
<required parameter>
[optional parameter]
```

The JiffyMON commands are broken down into three categories:

* General Purpose Commands
* Conversion/Arithmetic Commands
* Disk Drive Commands

General Purpose Commands

These are the commands that you will probably use most often, and perform functions common to many other machine language monitors. They include commands to display and alter bytes in memory, assemble/disassemble machine code, etc.

Using the General Purpose Commands with a Disk Drive

One of the most powerful features of JiffyMON is its ability to access disk drive memory for many of its commands. The JiffyMON commands available for drive access are listed below and are also indicated as such by the MODE heading at the top of applicable command descriptions.

General Purpose Commands Usable with a Disk Drive:

ASSEMBLE
COMPARE MEMORY
DISASSEMBLE
FILL MEMORY
GO
HUNT MEMORY
MEMORY DISPLAY
TRANSFER MEMORY

To use a particular command to access drive memory instead of computer memory is very simple. All you have to do is to include an asterisk (*) after the command letter of the desired command and then enter the rest of the command normally. For example, to use the DISPLAY MEMORY command to access the disk drive rather than the computer, add an asterisk immediately following the **M** in the command syntax:

M*0300 0320

The resulting memory display will reflect the contents of drive memory, rather than computer memory.

To help you keep track of where the information on your screen originated, all disk drive command outputs include an asterisk and/or address displayed in reverse video, as in the following MEMORY DISPLAY:

M*0300 0320

```
:*0300 00 00 00 00 00 00 00 00 ........
:*0308 00 00 00 00 00 00 00 00 ........
:*0310 00 00 00 00 00 00 00 00 ........
:*0318 00 00 00 00 00 00 00 00 ........
:*0320 00 00 00 00 00 00 00 00 ........
```

Conversion/Arithmetic Commands

These commands are included to help you convert between hexadecimal/decimal/binary values; convert from PETSCII, graphics, or control characters; do a memory checksum; and to perform arithmetic operations (+,-) on hexadecimal values.

Disk Drive Commands

Another feature of JiffyMON is its inclusion of a series of drive-specific commands. These commands enable you to transfer memory between computer and drive, and to read, alter, and write sectors to disk.

Conflicts with JiffyDOS Commands

There are two JiffyDOS commands that require special attention because of conflict with JiffyMON commands. These are the JiffyDOS **load machine language program** % command - which is the same as the JiffyMON **BINARY CONVERSION** command; and the JiffyDOS **LOAD ML (no warmstart) &** command - which is the same as the JiffyMON **CHECKSUM MEMORY** command.

The % command can be used both ways. If you enter % without a quotation mark (i.e. **%11111111**) it is the JiffyMON **BINARY CONVERSION** command. If you enter % with a quotation mark (i.e. **%"filename**), it is the JiffyDOS **load machine language program** command.

The **&** command can only be used as the JiffyMON **CHECKSUM MEMORY** command at the present time. Note that this restriction only applies to BASIC direct mode - within BASIC programs, the **&** command still operates as a JiffyDOS command.

General Purpose Commands

ASSEMBLER

MODE: Computer
Disk Drive

SYNTAX: **A <address> <opcode mnemonic> <operand>**

Starting the assembler requires the entry of the letter **A** followed by a starting address in hexadecimal. Following the address, you can enter a valid 6510 opcode mnemonic. The line is entered by pressing **RETURN.** If you enter an invalid mnemonic, a "**?**" will be displayed to indicate the error.

Once you have input one line, the assembler outputs the command letter **A** followed by the address of the next instruction. You can then enter the next instruction. Operand data and addresses are assumed to be in hexadecimal. For convenience, OPERAND DATA AND ADDRESSES DO NOT HAVE TO BE PREFIXED WITH **$** . Immediate data must be prefixed with **#** .

To exit the assembler, hit the **RETURN** key after the address prompt.

EXAMPLES:

A C700 LDA #21

JiffyMON will display:
A C700 A9 21 LDA #$21
A C702

A*0300 LDA (32),Y

Assembles code in disk drive RAM
JiffyMON will display:
A*0300 B1 32 LDA ($32),Y
A*0302

BREAKPOINT SET

MODE: Computer

SYNTAX: **B <breakpoint address> [count]**

This command enables you to set up a condition that will break (stop) execution of code after the particular instruction at the breakpoint address has been executed a specified number of times. For the break to occur, you must start code execution with the QUICK TRACE command. If no count is specified, the break occurs at the first occurrence of the specified address.

EXAMPLES:

B C080 000A

Break execution after the instruction at **$C080** has been executed 10 (**$000A**) times

B C080

Break execution the first time the instruction at **$C080** is executed

COMPARE MEMORY

MODE: Computer
Disk Drive

SYNTAX: **C <start address 1> <end address 1> <start address 2>**

This command compares two memory blocks. The start and end addresses of block 1 must be specified, along with the start address of block 2. Mismatching bytes are indicated by the output of the address where the mismatch is located. Compare can be stopped by hitting the **RUN/STOP** key during output of mismatched data location addresses.

EXAMPLES:

C C700 C7FF C800

Compare the block of memory at **$C700**–**$C7FF** to the block of memory starting at **$C800**

C*0300 03FF 0400

Compare the block of memory in dsk drv RAM at **$0300**–**$03FF** to the block of memory starting at **$0400**

DISASSEMBLER

MODE: Computer
Disk Drive

SYNTAX: **D <start address> [end address]**

This command enables you to view the disassembly of the memory block specified by the start and end addresses. If only the start address is specified (the end address is optional), then only the one instruction at that address is disassembled. Long running disassemblies can be halted by pressing the **RUN/STOP** key. Each disassembled line is output with a leading comma, an address, the bytes that make up the instruction, the opcode, and its operand/address, as in:

```
, C700 4C 00 C8 JMP $C800
```

Scrolling

The disassembly produced by this command can be scrolled up or down using the cursor keys to extend the address range beyond that which is shown on the screen.

Editing

Once disassembled, each instruction can be edited in two ways:

1. By moving the cursor up to the line that you wish to modify, and typing **A** over the comma. You can then enter new instructions starting at that point (see the **ASSEMBLER** command).
2. By moving the cursor up to the bytes that make up the instruction (**4C 00 C8** in the example above), and modifying those byte values. Enter the changes you make by pressing **RETURN.** The disassembly on that line will be updated according to the changes you have made. (The comma is a "hidden" command which you can use to enter hexadecimal data into memory with disassembly of the data as you enter it).

EXAMPLES:

D E000 E005 Disassembles code in the Kernal ROM

```
, E000 85 56    STA $56
, E002 20 0F BC JSR $BC0F
, E005 A5 61    LDA $61
```

D*E001 E003 Disassembles code in disk drive ROM

```
,*E001 A9 20    LDA #$20
,*E003 20 9D DD JSR $DD9D
```

EXIT JIFFYMON (cold start BASIC)

SYNTAX: **E**

This command disables JiffyMON and performs a cold start of BASIC. To restart JiffyMON after entering this command, enter **SYS 64715** (JiffyMON/64 2.0) .

FILL MEMORY

MODE: Computer
Disk Drive

SYNTAX: **F <start address> <end address> <byte value>**

This command fills a block of memory with the specified byte value. Note that the memory areas underneath the BASIC and Kernal ROMs and I/O Area cannot be filled unless the appropriate ROM (or VIC/SID/CIA1/CIA2/Color RAM) are switched out using one of the JiffyMON **SWITCH CONFIGURATION** commands.

EXAMPLES:

F 1000 1035 00 Fill memory block **$1000–$1035** with **00**
F 2000 20FF EA Fill memory block **$2000–$20FF** with **EA**
F*0300 03FF AA Fill memory block **$0300–$03FF** in disk drive RAM with **AA**

GO

MODE: Computer
Disk Drive

SYNTAX: **G [address]**

Computer Mode

This command starts execution of machine code at the specified (or default) address. The register image shown in the **REGISTER DISPLAY** command is loaded into the 6510 registers prior to execution of the code. If no address is specified, then execution begins at the address specified at the PC (program counter) register image location.

Disk Drive Mode

When this command is used to start execution of code in disk drive memory, a MEMORY EXECUTE (M-E) command is sent to the drive, telling it to start running the code at the address you specify. Note: When using this command to start a program in drive memory, this address must be specified each time (no default exists).

Halting Execution (Computer mode only)

There are two ways to halt the execution of code once it is started:

1. To implant a BRK instruction at the point you want the code to stop running.

2. To press **RESTORE** while the program you have started is running.

In either case, the program will halt, with the registers saved and displayed (unless, of course, your program has trashed JiffyMON). Also, the address of the instruction following the BRK is saved in PC for execution continuation later if another **GO** command is given.

EXAMPLES:

G 2000	Starts execution of the machine language code at address **$2000**
G	Starts execution at the address specified at the PC register image location (see the **REGISTER DISPLAY** command)
G* 0300	Starts execution of code at address **$0300** in dsk drv mem

HUNT MEMORY

MODE: Computer
Disk Drive

SYNTAX: **H <start address> <end address> <byte> [byte] ... [byte]**
H <start address> <end address> <'PETSCII CHARACTER STRING> (shift-7 => ')

This command scans a block of memory for a maximum of 32 characters or bytes of data. The address of each occurrence is printed out. During address output, the hunt can be stopped with the **RUN/STOP** key.

If the specified data is not found, no addresses are displayed, and the cursor will return to the screen.

EXAMPLES:

H 1000 2000 20 00 C7	Scans memory from **$1000** to **$2000** for the occurrence of the bytes **20 00 C7** (JSR $C700)
H C700 C800 'ABCDEF	Scans memory from **$C700** to **$C800** for the character string **ABCDEF**
H* 0300 06FF 'JIFFYMON	Scans memory from **$0300** to **$06FF** in the disk drive for the character string **JIFFYMON**

JUMP TO SUBROUTINE

MODE: Computer

SYNTAX: **J <address>**

This command calls a machine language subroutine, and then returns to the BASIC/JiffyMON environment when the RTS command for that subroutine is executed.

The register image shown in the **REGISTER DISPLAY** command is not used upon entry to the subroutine, and is not disturbed when the subroutine returns to JiffyMON.

EXAMPLE:

J C700	Calls the subroutine starting at address **$C700**

LOAD A FILE

MODE: Computer

SYNTAX: **L [address] <"filename"> [device]**

This command loads a file from disk or tape. The load address is an optional parameter – you can specify the address at which to load the file, or you can load the file at its default load address by not specifying an address. Also optional is the device number parameter. If you do not specify a device, JiffyMON defaults to the current JiffyDOS default device. (You can change the default device by pressing **CTRL-D**, or by using the JiffyDOS **@#** command – see your JiffyDOS manual).

For tape (device #1), if no filename is specified, the first file found is loaded.

The BASIC memory pointers are not affected by this load.

EXAMPLES:

L 4000 "TEST	Loads the file **TEST** from the current JiffyDOS default device starting at address **$4000**
L "TEST	Loads **TEST** at its default load address from the current JiffyDOS default device
L 2000 "TEST" 09	Loads **TEST** from device **#9** starting at address **$2000**
L "TEST" 01	Loads **TEST** from tape at its default load address

MEMORY DISPLAY

MODE: Computer
Disk Drive

SYNTAX: **M <start address> [end address]**

This command enables you to view memory in eight-byte segments followed by a PETSCII translation. The memory block that you wish to view is specified by the start and end addresses. If only the start address is specified (the end address is optional). then only the eight bytes starting at that address are displayed. Long running memory displays can be halted by pressing the **RUN/STOP** key. Each line is output with a leading colon, an address, eight byte values, and the PETSCII translation of those eight bytes, as in:

: C700 41 42 43 44 45 46 47 48 ABCDEFGH

Scrolling

The memory display produced by this command can be scrolled up or down using the cursor keys to extend the address range beyond that which is shown on the screen.

Editing

Once displayed, each line can be edited by moving the cursor up to the bytes that you wish to change, and modifying those byte values. Enter the changes you make by pressing **RETURN.** The PETSCII display on that line will be updated according to the changes you have made. (The colon is a "hidden" command which you can use to enter hexadecimal byte values into memory).

MEMORY DISPLAY (continued)

EXAMPLES:

M E47C E48B

```
: E47C 4A 49 46 46 59 44 4F 53 JIFFYDOS
: E484 20 56 36 2E 30 31 20 28  V6.01 (
```

M E4A0

```
: E4A0 42 41 53 49 43 20 56 32 BASIC V2
```

M*E5B8 E5C0 (displays drive memory)

```
:*E5B8 49 46 46 59 44 4F 53 20 IFFYDOS
:*E5C0 35 2E 30 20 31 35 34 B1 5.0 154Q
```

NEW LOCATOR

MODE: Computer

SYNTAX: **N <start address> <end address> <offset> <range low> <range high> [W]**

This command adjusts all three-byte instructions in the block defined by the start and end addresses by adding the offset value to the absolute address in the two bytes following the opcode. Any absolute addresses outside of the range are not adjusted. If an illegal opcode is encountered, the process stops with a disassembly and display of the bad opcode. The optional **W** parameter initiates a search for two-byte (word) addresses.

EXAMPLES:

N C700 C760 3000 CE00 CEFF	Adjusts three-byte instructions found within the range **$C700-$C760 ;** Addresses lying within the range **$CE00-$CEFF** are increased by **$3000 ;** Addresses outside of the **$CE00-$CEFF** range are not modified
N C000 C0FF 1080 C800 CFFF W	Adjusts two-byte word addresses found within the range **$C000-$C0FF ;** Word addresses within the range **$C800-$CFFF** are increased by **$1080 ;** Addresses outside of the **$C800-$CFFF** range are not modified

OFFSET CALCULATE

SYNTAX: **O <opcode address> <target address>**

This command calculates the offset for branch instructions. The first address is for the location containing the branch opcode, and the second address is the branch target address. The resulting offset byte is displayed in hexadecimal on the same line as the command.

EXAMPLE:

O C880 C820 9E	Displays the hexadecimal offset ($9E) required for an instruction branching from **$C880** to a target address of **$C820**

QUICK TRACE

MODE: Computer

SYNTAX: **Q [address]**

This command executes each instruction up to the point defined by the **BREAKPOINT SET** command. The address specified in the **BREAKPOINT SET** command is checked for the break on the Nth occurrence. Execution does not occur at full speed, as JiffyMON must check for the breakpoint after each instruction. Pressing **RUN/STOP** halts execution, and displays the register image.

EXAMPLES:

Q	Begins tracing at the PC register image location
Q 2000	Begins the trace at address **$2000**

REGISTER DISPLAY

MODE: Computer

SYNTAX: **R**

This command displays the 6510 register image saved each time a BRK instruction is executed. This image can be modified by typing over the displayed register values. The changes are entered into the image when you press the **RETURN** key.

EXAMPLE:

```
R
   PC  IRQ  SR AC XR YR SP
; E71C FC80 23 00 10 00 E9
```

SAVE MEMORY

MODE: Computer

SYNTAX: **S <start address> <end address+1> [relocation address] "filename" [device]**

This command saves the block of memory specified by the start and end addresses (note that the end address you specify must be the end address of the block +1). An optional relocation address may be specified. The block of memory will be saved to the current JiffyDOS default device, unless the optional device number parameter is specified.

EXAMPLES:

S C700 C900 "TEST	Saves memory block **$C700–$C8FF** to the current JiffyDOS default device
S C700 CA00 C000 "TEST	Saves memory block **$C700–$C9FF** to the current JiffyDOS default device; the file's default load address will be **$C000**
S C700 0800 "TEST" 09	Saves memory block **$C700–$C7FF** to device #9

TRANSFER MEMORY

MODE: Computer
Disk Drive

SYNTAX: **T <start address> <end address> <new start address>**

This command transfers (copies) a block of memory specified by the start and end addresses to a new location. Transfer begins at the high location of each block. This is an important consideration when the source and destination memory blocks overlap.

EXAMPLES:

T 1000 2000 3000	Transfers memory block **$1000–$2000** to a new location beginning at address **$3000**
T*0300 03FF 0400	Transfers memory block **$0300–$03FF** in the disk drive to a new location beginning at address **$0400**

SELECT CONFIGURATION

MODE: Computer

SYNTAX: **K**
B
I

SELECT CONFIGURATION is really three separate commands:

K switches the Kernal ROM		in/out (addresses $E000–$FFFF)
B switches the BASIC ROM		in/out (addresses $A000–$BFFF)
I switches the I/O Area	(VIC/SID/CIA1/CIA2/Color RAM)	in/out (addresses $D000–$DFFF)

Each of these commands is a toggle. That is, when you enter the command for the first time, it switches the specified ROM or I/O Area out of JiffyMON's visible address range. When you specify the command a second time, it switches the specified area in. Each time you enter one of these commands, a message (**IN** or **OUT**) is displayed, indicating the present status of the selected area.

When the ROMs or I/O Area are switched out, all JiffyMON commands will look at the hidden RAM in the specified area, allowing you to modify memory, assemble code, etc. in this RAM. The **GO** command will jump to the hidden RAM if the associated ROM is switched out (not true for the I/O Area). The **QUICK TRACE** and **WALK** commands will also step through the hidden RAM areas if the ROMs are switched out.

EXAMPLES:

K OUT	The Kernal ROM is switched out, letting you look at the RAM hidden underneath the ROM (be careful – this is where JiffyMON resides)
B IN	The BASIC ROM is switched back in
I OUT	The I/O Area (VIC/SID/CIA1/CIA2/Color RAM) is switched out, allowing you to look at the hidden RAM from $D000–$DFFF

VERIFY MEMORY

MODE: Computer

SYNTAX: **V [address] <"filename"> [device]**

This command verifies a file against a block of memory specified by a starting address. If no address is specified, then the block of memory is determined by the default load address of the file. If no device number is specified, the verification defaults to the current JiffyDOS default device.

EXAMPLES:

V "TEST	Verifies the file **TEST** against the block of memory residing at **TEST**'s default load address
V C700 "TEST	Verifies the file **TEST** against the block of memory starting at location **$C700**
V C800 "TEST" 09	Verifies the file **TEST** stored on device **#9** against the block of memory starting at location **$C800**

WALK

MODE: Computer

SYNTAX: **W [address]**

This command begins by initializing the 6510 registers to the register image shown in the **REGISTER DISPLAY** command. One instruction is executed, an IRQ is generated, and the new register image is saved and displayed along with the next instruction to be executed.

WALK can be continued by pressing any key except the **RUN/STOP** key and the **J** key.

Pressing the **RUN/STOP** key stops the **WALK** command. Pressing the **J** key while walking finishes execution of a subroutine at full speed. **J** can be pressed either at the beginning or in the middle of a subroutine. In either case, **WALK** resumes on return from the subroutine. Hitting any other key during walking causes execution of the next instruction. Caution: Hitting **J** when you are walking in mainline code will probably cause unpredictable results.

EXAMPLES:

W	Begin WALKing at the PC register image location
W 2000	Begin WALKing at address **$2000**

EXIT JIFFYMON (warm start BASIC)

SYNTAX: **X**

This command disables JiffyMON and performs a warm start of BASIC. BASIC programs in memory are not affected. To restart JiffyMON after entering this command, enter
SYS 64715 (JiffyMON/64 2.0) .

Conversion/Arithmetic Commands

HEXADECIMAL CONVERSION

$4142 16706 A B 0100 0001 0100 0010

A hexadecimal number is input to obtain the decimal, PETSCII characters for the two bytes, and binary values. The values produced by this command can be scrolled up or down using the cursor keys to get decreasing/increasing numbers converted beyond that which is shown on the screen.

DECIMAL CONVERSION

#16706 4142 A B 0100 0001 0100 0010

A decimal number is input to obtain the hexadecimal, PETSCII characters of the two bytes, and binary values.

BINARY CONVERSION

%0100000101000010 4142 16706 A B

A binary number is input to obtain the hexadecimal, decimal, and PETSCII characters of the two bytes.

PETSCII CONVERSION

"B 42 66 0100 0010

A PETSCII, graphics, or control character is input to obtain the hexadecimal, decimal, and binary values.

CHECKSUM MEMORY

& C000 CFFF A500

The data for the memory block **$C000**–**$CFFF** inclusive is byte-summed and displayed.

ADDITION

+1111 2222 3333

Two hexadecimal numbers are input to obtain their modulo 16 ($10) sum.

SUBTRACTION

–3333 1111 2222

Two hexadecimal numbers are input, and the second is subtracted from the first to obtain their difference. Subtraction is done with two's complement arithmetic.

Disk Drive Commands

JiffyMON offers the following commands which allow you to perform memory transfers between computer and disk drive, and to read/write sectors to and from disk.

Each command references the current JiffyDOS default device (disk drive). To switch the drive default before entering one of these commands, press **CTRL-D** or use the JiffyDOS **@#** command (see your JiffyDOS manual).

BLOCK READ

SYNTAX: **BR [track] [sector] [computer address]**

This command reads the specified sector from disk and transfers its contents (256 bytes) to a specified area in the RAM of your computer. Track and sector values are in hexadecimal. If you do not specify a track and sector, the last sector referenced in a **BR**, **BRL**, **BR+**, or **BW** command is used. If you do not specify a destination address, the sector is transferred to the address last used during a **BR**, **BRL**, **BR+**, or **BW** command.

If the block read is successful, the first $B8 bytes of the sector will be displayed as in the **MEMORY DISPLAY** command. The sector contents can then be edited or redisplayed by any applicable JiffyMON command. The cursor keys can be used to scroll through the remainder of the sector bytes. Errors are indicated by a flashing drive error light. Use the JiffyDOS **@** command to display the error if one occurs.

EXAMPLE:

BR 12 00 C800	Reads the contents of Track 18 (**$12**) , Sector **0** to computer address **$C800**

BLOCK READ NEXT SECTOR ACCORDING TO TRACK AND SECTOR LINK

SYNTAX: **BRL [address]**

This command reads the next sector on disk according to the track and sector link of a file. The track and sector link is obtained from the last sector referenced by a **BR**, **BRL**, **BR+**, or **BW** command. The contents of the sector is transferred to computer RAM at the specified address. If no address is specified, the command defaults to the last address used by a **BR**, **BRL**, **BR+**, or **BW** command.

The display and editing of the sector on screen works identically to the **BLOCK READ** command.

EXAMPLE:

BRL C900	Reads the next track and sector of a file to address **$C900** in computer RAM

BLOCK READ NEXT SECTOR ON DISK

SYNTAX: **BR+ [address]**

This command reads the next sector on disk (i.e. Track 01, Sector 01 will be read if the prior sector read was Track 01, Sector 00). The prior track and sector is obtained from the last sector referenced in a **BR**, **BRL**, **BR+**, or **BW** command. The contents of the sector is transferred to computer RAM at the specified address. If no address is specified, the command defaults to the last address used by a **BR**, **BRL**, **BR+**, or **BW** command.

The display and editing of the sector on screen works identically to the BLOCK READ command.

EXAMPLE:

BR+ C900	Reads the next sector to address **$C900** in computer RAM

BLOCK WRITE

SYNTAX: **BW [track] [sector] [computer address]**

This command writes 256 bytes from computer RAM to the specified track and sector on disk. The base address of the 256 bytes written can be specified in the command, or can be left out – in which case the **BW** command will default to the last address used in a **BR**, **BRL**, **BR+**, or **BW** command. Specifying the track and sector for **BLOCK WRITE** is also optional. When specifying a track and sector, they must be in hexadecimal. If no track and sector are specified, the **BW** command will default to the last track and sector used in a **BR**, **BRL**, **BR+**, or **BW** command.

EXAMPLES:

BW 20 10 C700	Writes 256 bytes from computer address **$C700** to track 32 (**$20**) , sector 16 (**$10**)
BW 10 0A	Writes 256 bytes from the address last referenced in a **BR, BRL, BR+,** or **BW** command to track 16 (**$10**) , sector 10 (**$0A**)
BW	Writes 256 bytes to disk according to the last computer addr. and track/sector used in a **BR, BRL, BR+,** or **BW** command

MEMORY READ

SYNTAX: **MR <start address> <end address> <computer address>**

This command transfers a block of memory from disk drive to computer RAM. The start and end addresses of the memory block in the disk drive are specified along with the starting address of the destination block in computer RAM.

EXAMPLE:

MR 0300 03FF 1000	Moves the contents of memory block **$0300–$03FF** in the disk drive RAM to computer RAM starting at address **$1000**

MEMORY WRITE

SYNTAX: **MW <start address> <end address> <drive address>**

This command transfers a block of memory from your computer to disk drive RAM. The start and end addresses of the block of memory in the computer are specified along with the starting address of the destination block in disk drive RAM.

EXAMPLE:

MW 1000 10FF 0300	Moves the contents of memory block **$1000–$10FF** to disk drive RAM starting at address **$0300**

SECTION 4
JiffyMON COMMAND SUMMARY

General Purpose Commands

ASSEMBLER	A <address> <opcode mnemonic> <operand>
BREAKPOINT SET	B <breakpoint address> [count]
COMPARE MEMORY	C <start address 1> <end address 1> <start address 2>
DISASSEMBLER	D <start address> [end address]
EXIT (COLD START BASIC)	E
FILL MEMORY	F <start address> <end address> <byte value>
GO	G [address]
HUNT MEMORY	H <start address> <end address> <byte> [byte] **....** [byte] H <start address> <end address> <'PETSCII character string> (shft-7 ')
JUMP TO SUBROUTINE	J <address>
LOAD A FILE	L [address] <"filename"> [device]
MEMORY DISPLAY	M <start address> [end address]
NEW LOCATOR	N <start address> <end address> <offset> <range low> <range high> [W]
OFFSET CALCULATE	O <opcode address> <target address>
QUICK TRACE	Q [address]
REGISTER DISPLAY	R
RESTART	SYS 64715 (JiffyMON/64 2.0)
SAVE MEMORY	S <start address> <end address+1> [relocation address] "filename" [device]
TRANSFER MEMORY	T <start address> <end address> <new start address>
SELECT CONFIGURATION	
(Kernal ROM)	K
(BASIC ROM)	B
(I/O Area)	I
VERIFY MEMORY	V [address] <"filename"> [device]
WALK	W [address]
EXIT (WARM START BASIC)	X

Conversion/Arithmetic Commands

HEX CONVERSION	$<hexadecimal number>
DECIMAL CONVERSION	#<decimal number>
BINARY CONVERSION	%<binary number>
PETSCII CONVERSION	"<PETSCII character>
CHECKSUM MEMORY	& <start address> <end address>
ADDITION	+ <4-digit hex number> <4-digit hex number>
SUBTRACTION	- <4-digit hex number> <4-digit hex number>

Disk Commands

BLOCK READ	BR [track] [sector] [computer address]
BLOCK READ NEXT LINK	BRL [computer address]
BLOCK READ NEXT SECTOR	BR+ [computer address]
BLOCK WRITE	BW [track] [sector] [computer address]
MEMORY READ	MR <start address> <end address> <computer address>
MEMORY WRITE	MW <start address> <end address> <drive address>