CSC [14]6805 AND 6502 SIMULATORS

PROGRAM DEVELOPMENT AND DEBUGGING TOOLS

by Edgar M. (Bud) Pass, Ph.D.
Computer Systems Consultants, Inc.
1454 Latta Lane, Conyers, GA 30207
Telephone Number 404-483-1717/4570

INTRODUCTION

The [14]6805 and 6502 simulators are programs which enable the user to simulate, examine, and/or modify object [14]6805 and 6502 program files on disk or in memory on 6800 systems running under FLEX* or 6809 systems running under FLEX* or OS/9* or UNIFLEX*. Programs may be disassembled into source code format and the source may be displayed or printed. The programs comprising the [14]6805 and 6502 simulators are supplied in source form on disk - assembly is required. The systems are available on 5" or 8" diskette for FLEX* or OS/9* or UNIFLEX*. They are priced and marketed separately; however, because they are such similar systems, the summaries and manuals are combined. A detailed operating manual is provided, in addition to this summary.

THE [14]6805 AND 6502 SIMULATORS

The simulators have been designed to run on the 6800 or 6809 Motorola microprocessors under FLEX*, UNIFLEX*, and OS/9* operating systems. Under FLEX*, when assembled and run on a 6809 processor, they make the necessary address changes to run under FLEX* 9. The [14]6805 or 6502 program they analyze may reside on a diskette or (under FLEX*) in memory. The output files they produce may be sent to a disk file, printer, or terminal. The simulators are essentially self-instructive and have two levels of help files, in addition to a printed manual.

Each simulator has been designed for session-oriented use. At the beginning of the session, all indicators are reset to their default values, as indicated below. The user may then issue various commands. All simulator commands are single letters. If any other input is found when a command is expected, the command menu is displayed. Because of the interdependence of many of the commands, they should be issued in the proper sequence and at the proper times. The explanation below is intended to serve as a quick guide to elementary operation of the [14]6805 and 6502 simulators. A more complete explanation may be found in the printed manual.

The 'Z' command is used with either simulator. For the [14]6805, it indicates whether the processor being simulated is the 6805 (HMOS) version or the 146805 (CMOS) version. The STOP and WAIT instructions are implemented on the 146805 but not on the 6805. By default, the largest 146805 (16K bytes memory, 63 bytes stack) with STOP and WAIT instructions using 1 Mhz clock and external timer is assumed, except under OS/9, in which the amount of memory is restricted by the data space available to the OS/9 task below the simulator's stack. The [14]6805 external clock rate may be set by the 'Z' command.

The range of memory addresses being simulated may be specified or restricted by the 'Z' command. This range is used also to determine the location of the interrupt vectors. Memory access may be restricted thru the 'P' command, described below.

The amount of memory or stack on the [14]6805 or 6502 may be specified thru the use of the memory protection commands. The amount and range of memory available for simulation is limited by the necessity of the simulated object program being co-resident in memory with the simulator and the 6809 operating system or the UNIFLEX* stack and other overhead in the user's space. OS/9 memory allocation may be changed with an OS/9* command line parameter. Specific features of certain CPU models, such as A/D conversion, on-board parallel and serial ports, etc. are not simulated directly but may be handled with user code in the simulator. The [14]6805 timer is simulated, to the degree possible.

The user may need to determine the current indicator settings and control table contents. The 'L' command may be used at any time to list the indicators and tables.

If the user desires to process a program currently on disk, the 'S' command may be used. 'S' will prompt for an input file name. Since the 'S' command resets most indicators and tables and the simulated [14]6805 or 6502 program in memory, it should normally be issued only near the beginning of a session. If 'S' is not issued, or 'S' is issued but no file name is entered, then the [14]6805 simulator assumes that the object program resides in memory, which is meaningful only under FLEX*, and not under OS/9* and UNIFLEX*. If 'S' is issued, the object program is loaded for simulated execution.

One command which should be used before an 'S' command, if desired, is the 'O' command. This command provides an offset value which is added to each address in the program being processed. If the program is to be processed from disk, the offset value is applied when the program is loaded. The 'O' command has no effect if the program is to be processed directly from memory.

If the program is being processed from disk, the starting and ending addresses will be set automatically. The transfer address will be set if it is present in the file. In any case, the 'N' command may be used to set or to change the start, end and transfer addresses. If the transfer address is set to FFFF, no transfer address will be generated in the output file. If an output is attempted and no start, end, or transfer address has been provided, they will be requested. The transfer address is also used to set the simulated initial program counter address.

At any point after start and end addresses have been defined, the 'Q' command may be used to format and display the program on the terminal. The terminal must display at least 80 columns and 18 rows. Each page of the display shows 256 bytes of the program. The first page of the display begins at the address represented by the starting address with the low-order byte zeroed. Then the display may be paged forward, backward, set to an arbitrary 256-byte sector of memory, or terminated. Each page of the display may also be modified in a full-screen edit manner. Data may be entered in either hexadecimal or alpha format depending upon the area of the screen to which the cursor is pointing. The displayed data represents the true resolved contents of the program in main memory or on disk. If the input program file is composed of multiple redefinitions or the 'Q' or 'M' command or simulated execution of the [14]6805 or 6502 program code changed the value of the byte at a given address, only the last definition of a particular byte will be displayed.

The 'V' command is used to request a listing of the program code between the then-defined starting and ending addresses. This listing is produced in instruction, FCB, FCC, and FDB formats. The readability of the listed program code may be improved substantially in many cases thru the 'typing' of memory ranges. The 'Q' command may be used to help determine how to split memory into contigous ranges of instructions, constants, ignored areas, etc., if a source listing is not available.

The 'D' and 'E' commands are used to initiate simulated execution of the object program. If the 'D' command is used, the contents of the simulated registers are shown before execution of each instruction and the simulator waits for a key to be struck before continuing execution. If CNTRL-I is struck, the external interrupt bit is set in the simulated condition code register. If CNTRL-R is struck, a simulated reset operation is performed before execution of the next object instruction. Both commands cause execution to continue until a CNTRL-C is struck or a breakpoint, protection exception, STOP/WAIT, or invalid instruction is encountered. The setting of the printer switch, which is controlled by the 'M P' and 'M N' commands, determines whether the output will be sent both to the terminal and printer or terminal only, in response to a 'D' command.

The simulated registers may be displayed and optionally modified thru the use of the 'R' command. This command first displays the register names and the register contents. Then the cursor is placed over the first position of the first register. Using the left and right cursor control keys, the user can reposition the cursor to a desired register location and modify it by over-typing it. Illegal values may not be entered by this method. After the modifications have been made, the command is terminated with a carriage return.

Breakpoints may be set with the 'B' command and reset with the 'X' command. All addresses in the specified range have breakpoints inserted or removed. The object memory is not modified, as the breakpoint information is carried separately, so entire ranges of memory may have breakpoints set or reset without loss of data. All bytes of a multi-byte instruction are checked for the presence of a breakpoint. However, the presence of a breakpoint is ignored on an instruction which was not executed because of a breakpoint, or when using the 'D' command, which is equivalent to a breakpoint on every instruction.

Memory protection may be controlled thru the use of the 'P' command. Ranges of memory addresses may have the following attributes set or reset:

A	reset memory protection
E	access-protect (execute-only)
M	memory-protect (not present)
N	execute-protect (non-executable)
R	write-protect (read-only)

All memory addresses outside the range of $0000 thru $3FFF (for the

[14]6805) or within the range of the simulator (for the 6502) are automatically memory-protected (M). Areas of memory outside the bounds of the program and below address $000B (for the [14]6805) are non-executable (N), as are those "typed" as non-instruction. As in the case of the breakpoint commands, object memory is not actually modified by these commands.

The 'T' command may be used to output the addresses of the last 255 changes of control (branches, calls, returns, and interrupts) executed by the simulated program. This is often useful in debugging a program which has taken an unexpected branch since it may show the path taken by the program counter before it took the unexpected branch. The addresses are output in reverse order, 15 addresses per line. The table is reset each time that a 'D' or 'E' command is issued.

Under FLEX*, if a printer driver name is provided when the simulator is executed, the displayed output may also be sent to the printer, in addition to the display, when the 'D', 'E' and 'V' commands are specified. This output is controlled by the printer switch, which may be set and reset with the 'M P' and 'M N' commands.

Once an area of memory has been 'typed', the following commands may be used to improve the output display and printed formats:

```
     A-FDB address range
     C-FCC address range
     H-FCB address range
     I-instruction address range
     J-instruction+ASCII address range
     K-ignored address range
```

Each command above will request a memory range for the given type of memory. The last definition of a given byte is used in each case.

The 'M' command may be used to examine and change object memory and certain simulator parameters. When an 'M' command is entered, a starting address is requested. If a four hex-digit address is entered, the object code byte at that location will be displayed. If the user desires to change the byte to another value, the new value may be entered. In this case, and in most other cases, the byte at the next location is displayed. If '^' is entered, the byte at the previous location is displayed. If carriage-return is entered, the command is terminated. If 'N', 'P', or 'T' is entered when an address is requested, the input is interpreted as a simulator subcommand. The 'P' and 'N' subcommands toggle the printer switch on and off, respectively; in order to enter these subcommands, the printer driver name must precede the name of the simulator on the FLEX* command line. The 'T' subcommand may be used to fill an entire range of program addresses with the same one-byte hex string. When this subcommand is entered, the starting and ending addresses and one-byte hex string are requested.

The 'Y' command is used to scan for a hex string of bytes between a given range of addresses. The beginning addresses of the matching strings are printed in response. When the 'Y' command is entered, the starting and ending addresses and matching string are requested. A carriage return may be used to terminate the matching string.

The 'W' command is used to output the resulting object program to a disk file. An output file name is requested. If the start and end addresses have not been provided, they are requested. A non-OS/9 output file reflects only that program code between the start and end addresses, exclusive of ignored address ranges. Ignored address ranges may be generated explicitly (thru the 'K' command) or implicitly by not being defined in an input file. All changes made explicitly (thru the 'M' and 'Q' commands) and implicitly (thru the execution of the simulated program) are reflected in the output file. Under FLEX* or UNIFLEX*, after the revised program code has been written to the output file, the current transfer address is output if it is not equal to FFFF.

Although most of the commands are simple, repeated entry of a large number of them may become tedious and time-consuming, especially when a large program is being debugged in an iterative fashion. The 'G' command allows the user to store commands such as 'A', 'C', 'H', 'I', 'J', 'K', 'M' in a text file and input them later to the simulator. It may be used whenever the '?' prompt is displayed. Any errors detected in the input text file cause the immediate termination of the reading of the file and return control to the terminal. The state of the simulator at a given time may thus be saved for later use. The state of the object program is not saved; this includes changes to memory made by simulated execution and includes breakpoint setting and clearing commands.

The 'U' command may be used to execute a FLEX* or OS/9* or UNIFLEX* command while still in the [14]6805 or 6502 simulator. Any FLEX* command which does not interfere with the memory space occupied by the simulator or object program may be used. No check is made to determine whether a command is valid.

The 'F' command may be used to return to FLEX* or OS/9* or UNIFLEX*.

GENERAL

For ease of customizing the simulator in situations other than for FLEX* 2 or FLEX* 9, and to support various terminals and computers, all external addresses are maintained in an area near the beginning of the program and cursor command characters are kept in tables which may be easily modified at assembly time.

* FLEX and UNIFLEX are trademarks of Technical Systems Consultants.
* OS/9 is a trademark of Microware.

COMMAND SUMMARY

Address range commands:

```
 A-FDB,C-FCC,H-FCB,I-code,J-code+ASCII,K-ignored
 B/X-set/reset breakpoint
 P-change protection attributes
```

Operational commands:

```
 D/E-execute with/without display
 M-modify memory or parameters
 Q-query object code
 R-display/modify simulated registers
 T-display trace table
 U-enter FLEX or OS/9 or UNIFLEX command
 V-view object code
 W-write new object code file
 Y-find hex string in object code
```

Miscellaneous commands:

```
 F-exit to FLEX or OS/9 or UNIFLEX
 G-specify auxiliary input/output file
 L-list control information
 N-set new memory range
 O-set offset load value
 S-load input file to memory
 Z-change CPU model
```