CENTURION

CPU-6

DEALER SUPPORT MANUAL

DEBUG UTILITY

November 29, 1979

Centurion Computer Corporation
Richardson, Texas

TABLE OF CONTENTS

INTRODUCTION

The DEBUG utility is designed to aid in the debugging of CPL and Assembly language programs for the CPU6. The major difference between the old BUG and new DEBUG is that the new DEBUG utility is limited in access to the partition memory in which it is running, and will not impact other active partitions. The utility resides in an "E" type file name "@DEBUG", which must be resident on the system disk to be used.

The utility operates by being called in the .RUN statement of the program to be debugged. If the keyword (DBUG) is found, a special transient loads first the main program with the added offset factor specified (if any), and then loads the DEBUG program. Control is then transferred to the DEBUG program, which contains the main program entry address in the 'P' register.

USAGE and COMMANDS

The DEBUG utility is evoked by the use of the keyword "(DBUG)" or "(DBUG,X)" in the .RUN statement for the program. If "X" is specified, an offset factor of from 1 to 9 kilo-bytes is added between the end of the program and its associated externals (HICORE), and the beginning of the DEBUG utility.

EXAMPLE:

```
.RUN XABC (DBUG,4)
```

When control is passed to the DEBUG utility, memory may be examined or modified in either hexidecimal or decimal notation, and breakpoints (traps) may be set.

While the DEBUG utility has control, there are a number of commands available. These commands are explained in the following pages.

COMMANDS

Oxxxx SET OFFSET

Usually the offset value (not to be confused with the offset factor discussed previously) is set at the beginning of the partition, X'8000', by the utility. This is not changed unless it is necessary to debug an external subroutine or overlay, since these are assembled separately and have their own offset values. When it is used, the offset specified by xxxx will be added to all memory addresses entered.

Gxxxx,yyyy,zzzz GOTO

This command has several forms. It basically is a transfer of control command, which, depending on the form, can restart the program either at the entry point, or various points in the program. The possible forms are:

G	restart (continue) program from last trap.
Gxxxx	start program execution at hexidecimal address xxxx.
G,yyyy	restart program at entry point, and set one trap at hexidecimal address yyyy.
G,yyyy,zzzz	restart program at entry point, and set two traps at hexidecimal addresses yyyy and zzzz.
Gxxxx,yyyy	start program at hexidecimal address xxxx, and set one trap at hexidecimal address yyyy.
Gxxxx,yyyy,zzzz	start program execution at hexidecimal address xxxx, and set two traps at hexidecimal addresses yyyy and zzzz.

NOTES on TRAPS: A trap is set by placing the assembly language instruction "JSR/ DEBUG" in a specified location, and saving the program instruction(s) at that location. All traps are automatically cleared when control is passed to the DEBUG monitor. The instruction(s) at the trap location is restored in memory, but has not yet been executed.

The trap instruction (JSR/ DEBUG) requires three bytes, thus the following:

Traps should not be placed within 3 bytes of one another.

A trap should not be placed less than 3 bytes before a label that may be executed before the trap.

A trap should not be placed less than 3 bytes before the current GOTO address.

Traps should always begin on an instruction boundary.

Qxxx Quit

The Quit command causes a CPL STOP command to be executed. The completion code will be set to the decimal value entered, if a value is not entered, the completion code will remain unchanged.

H(expr) HEX/DECIMAL CALCULATOR

This command is used to perform calculations in either hexidecimal or decimal. Legal functions are addition (+), subtraction (-), division (/), or multiplication (*). Functions are evaluated from left to right, and results will be displayed in decimal notation.

Fxxxx,yyyy,zz FILL

The FILL command flls the memory locations from the offset hexidecimal address xxxx through yyyy with the hexidecimal or decimal value of zz.

Dxxxx,yyyy DUMP

The DUMP command will dump the contents of memory for offset hexidecimal address xxxx through yyyy to the device assigned to SYSLOG. The dump is displayed using the standard dump format, and is handled by the dump transient.

(N)xxxx Display/Modify (N)-byte integer

The integer display command, where (N) represents a 6, 4, or 2-byte integer, displays in decimal the integer that begins at offset hexidecimal address xxxx. Once the integer has been displayed, it may be modified by entering the new value in hexidecimal notation (beginning with an 'X'), or in decimal. If no value is entered, the displayed value will remain unchanged.

Mxxxx Display/Modify Memory .

The Memory display/modify command will display one hexidecimal byte, at the offset hexidecimal address xxxx. When the address is displayed, the contents of the memory location may be modified by entering a new hexidecimal value, or it may be left unchanged by entering a terminator (either a space, comma, or NEWLINE). Each of these terminators has a slightly different function. A space will display the contents of the next memory location, allowing modification; a comma opens the next memory location to modification without displaying the present contents; and the NEWLINE terminates the Memory display/modify function. Successive commas will cause the memory locations to be opened without changing the prior locations contents.

(R) Display/Modify Register

The Register command allows the display and modification of partition registers. When one of the characters V,A,B,X,Y,Z,S,C or P is entered, the current value of that register, in hexidecimal notation, is displayed. The contents of this register may then be modified by the entry of a new hexidecimal value, or it may be left unchanged by entering a terminator (space, comma, or NEWLINE). When the terminator is a space, the contents of the next register in the sequence (A,B,X,Y,Z,X,C,P) is displayed and available for modification. When the terminator is a comma, the next register becomes available for modification without being displayed. Successive commas will cause the consecutive registers to be opened for modification without changing the values of previous registers. The Register display/modification function is terminated either after the P register has been displayed/modified, or by a NEWLINE.

NOTE: The V register is the indicator register that stores the status of flags set previously in the program. These flags are the Fault, Link, Minus, and Value (also known as zero) and are represented as a hexidecimal value. All digits in the hexidecimal are set

initially to 0. If a flag should be set in the program, the value of that flag becomes 1. With the hexidecimal set up in the form of FLMV (representing the fault, link, minus, and value flags) the hexidecimal 0110 indicates that the link and minus flags are set and the fault and value are not.

CAUTIONS

The DEBUG utility is set up in a manner to prevent it from impacting other partitions, or from damaging any files except for those files being directly referenced in the program. There are no internal error messages; invalid commands are ignored. The only possible system error message that could be referenced is the AB-33 Program Malfunction.