GENERAL PRECISION, INC. / COMMERCIAL COMPUTER DIVISION

ABSTRACT

"Program Checkout 1," program K9-01.0, is a debugging package, i.e., a group of routines designed to help the programmer test and correct his program. This package embraces many of the functions previously performed separately by the following programs:

Memory Print 1, program K2-01.0
Memory Print 2, program K2-02.0
Address Search 1, program K3-01.0
Memory Clear 1, program K6-01.0
Change and Transfer 1, program J1-01.0

The package also provides additional debugging aids: program stops, print-out of changes made, and a choice of output formats.

"Program Checkout 1" consists of a number of subroutines, each of which operates independently when called on by one of 15 possible input codes.

Specifically, "Program Checkout 1" will perform the following functions:

1. Change (i.e., store) an instruction word.
2. Read in a relocatable decimal tape.
3. Print out a Table of Changes automatically after each 16 entries, or on demand of the program operator.
4. Print out the storage content of any area in memory.
5. Print out the object program in logical sequence, asking the operator at every test point which way to branch.
6. Insert temporary stops in the program for use during a subsequent checkout run.
7. Display the contents of a location in the Lower Accumulator.
8. Search any or all areas in memory for any specified bit configuration.
9. Clear a block of memory to any bit configuration desired.
10. Transfer control to a specified instruction in the object program.

STORAGE

MAIN MEMORY REQUIRED 13 tracks

RECIRCULATING SECTORS USED None.

STORAGE (Cont.)

INDEX REGISTER — Used in its full capacity by the subject program. It content prior to entry is not restored by the subject program.

UPPER AND LOWER ACCUMULATOR

Both are used by the subject program. Their contents prior to entry are not restored by the subject program.

Before "Program Checkout 1" is entered the Lower Accumulator must be in One Word Mode. It is left in that mode upon exit.

INPUT

15 input codes provide access to the various debugging routines in the program. The program will halt on an input order until the operator enters the desired code either from the keyboard or a previously prepared tape. The program assumes that the desired input-output devices have already been selected before an input code is given to it.

TABLE 1

Function	Code
Set modifier	Sddddd*
Change contents (fill memory)	ddddd*****
Print changes	PCxxxxx*
Print in instruction format (sequentially)	PIddddd*nnnnn*
Print in hexadecimal format (sequentially)	PHddddd*nnnnn*
Print in alphanumeric format (sequentially)	PAddddd*nnnnn*
Print as for PA plus 2 high order binary bits	PZddddd*nnnnn*
Follow object program print-out (logical sequence)	FPddddd*nnnnn*
Set address stop	SSddddd*
Remove address stop	RSddddd*
Remove all address stops	XSxxxxx*
Display in Lower Accumulator	Rddddd*
Search memory	SMddddd*nnnnn*
Clear memory	KMddddd*nnnnn*
Transfer	Tddddd*

INPUT (Cont.)

ddddd = initial address; must contain 5 characters except when a "NO LETTER" code precedes it.

xxxxx = meaningless address; must contain 5 characters.

nnnnn = final address; contains zero to 5 characters.

Note: If nnnnn is blank (for example PIdddddd**) the program assumes it is equal to ddddd, thereby limiting operations to one address. In all the input codes, the ddddd and nnnnn addresses are inclusive.

Each of the input codes is explained in detail on the following pages:

S CODE Example: S00400*

Set up a constant to modify the location, data-address, or next-address of any instruction containing an "M" code (see "NO LETTER CODE"). Any address preceded by an "M" code will be incremented by the value denoted by the "S" code. The constant must be in decimal format. The object program tape must also be in decimal format. The "S" code value remains constant until changed by another "S" code. If no address modification is desired, an "S" code followed by 5 zeros should be used. The "S" code must always be followed by 5 decimal digits.

NO LETTER CODE Example: 952*

This code has no letter preceding the initial address. Only the absolute address desired is used. The instruction word following the code is to be stored in this location. The content of the location is to be entered in the Table of Changes provided SENSE SWITCH 32 is raised.

To call on address modifier, precede 5 character address in word with an "M".

For example:

Modifier has been set at 400. The instruction word (RAU*405*407) is to be stored in location 952 as follows:

952*RAU*M00005*M00007**

INPUT

NO LETTER CODE (Cont.)

When 16 entries have been made using "NO LETTER" code the program will print out

"PRINT CHANGES"

whereupon the operator types in one of the following:

Response	Effect
YES*	The changes that have been made are printed in instruction and hexadecimal format. Table of Changes is then erased and program is ready for a new code.
NO* (or anything other than YES*)	Table of Changes is erased and program is ready for a new code.

PC CODE — Example: PC00000*

Print changes made with "NO LETTER" code thus far. Address is meaningless. Output of Table of Changes follows. One location and its content in the form of instruction and hexadecimal number is printed per line.

PI CODE — Example: PI12200*12263*

Print, in instruction word format, the contents of all locations from ddddd to nnnnn. Print one location and 3 instructions per line.

PH CODE — Example: PH12200*12203*

Print in hexadecimal format the contents of the locations from ddddd to nnnnn. Print one location and 3 hexadecimal words per line.

PA CODE — Example: PA01422*1437*

Print in alphanumeric format the 30 low order bits of the locations from ddddd to nnnnn. Each word shows the five, six-bit characters held in bit positions 2-31. Print one location and 3 alphanumeric words per line.

INPUT (Cont.)

PZ CODE — Example: PZ00722*737*

Print, as with "PA" code, the contents of locations from ddddd to nnnnn, but preceding the alphanumeric characters show the binary configuration of bit positions zero and 1. Print one location and 3 alphanumeric words per line.

FP CODE — Example: FP00000*3000*

Follow the object program from ddddd to nnnnn, i.e., print the instructions in logical sequence. After printing a "test" command (TBC or TMI) the program halts. The operator types in either d* (take D-address) or n* (take N-address) and printing continues along the path selected. Print one location and 3 instructions per line.

SS CODE — Example: SS00038*

Set address stop at this location. As many as 16 address stops may be set in a program at one time since a table of 16 halt instructions has been stored into memory by the subject program. An SS code exchanges one of the halt instructions and its location with the location and contents of the specified address. Depressing START COMPUTE button continues the program run.

RS CODE — Example: RS00038*

Remove the address stop from this location and replace with original contents. An error stop will result if an attempt is made to remove a stop that has not been set with "SS CODE."

XS CODE — Example: XS11111*

Remove all address stops from object program and replace them with original contents of the locations where the stops were inserted. Address is meaningless but must contain 5 characters.

R CODE — Example: R12706*

Display contents of this location in the Lower Accumulator.

INPUT (Cont.)

KM CODE — Example: KM00800*00962*

Clear memory from ddddd to nnnnn. Program inquires

"CONSTANT="

whereupon operator types, in hexadecimal format, the binary configurations desired. Clearing is completed when manual input light glows on typewriter and carriage returns.

SM CODE — Example: SM00000*12763*

Search memory from ddddd to nnnnn. Memory Search routine asks in turn for the argument, the mask, and the output desired. The operator types in responses. Response to argument query must be in instruction format. Response to mask query must be in hexadecimal format.

For example:

Operator wishes to search memory for hexadecimal word 00010001. Code and address are typed in as shown in above example. Then the following occurs:

Query	Response
ARGUMENT=	HEX*1*1**
MASK=	FFFFFFFF*
LOCATION ONLY	YES* (if only found location is desired). NO* (if found location and contents are desired). Any response other than YES* is considered to be NO*.

Finally the following is printed:

```
LOCATION ONLY YES*
12763

LOCATION ONLY NO*
12763 XSNS [X00] 00004 00000 [0000100001]
```

INPUT (Cont.)

T CODE Example: T01963

Transfer control to this location if SENSE SWITCH 8 is depressed. If SENSE SWITCH 8 is raised, stop and then transfer by depressing START COMPUTE button on computer.

CALLING SEQUENCE

The program is entered manually, by transferring control to sector zero of the first track in which the program is stored (See Royal McBee publication, RPC-4000 Programming Manual).

OUTPUT

Output consists of print-outs stating desired location and/or its content depending upon input code used. The format may be hexadecimal, instruction, or alphanumeric, again depending upon input code used.

OPERATING PROCEDURE

1. LOADING PROCEDURE

This program tape is in relocatable hexadecimal format and contains its own bootstrap routine. See ROAR, program H3-01.0, for loading procedure.

NOTE: SENSE SWITCH 1 options for relocating the tape have been provided as follows:

SENSE SWITCH 1 Depressed - Bootstrap routine will stop for input of track address to be used in relocating "Program Checkout 1."

SENSE SWITCH 1 Raised - Bootstrap routine will not stop for input of a modifier and will load "Program Checkout 1" starting in Track 0, Sector 0.

OPERATING PROCEDURE (Cont.)

2. INPUT-OUTPUT DEVICES REQUIRED

Selections must be made by operator.

3. SENSE SWITCH OPTIONS

Sense Switch	Function
1	RAISED - No Stop. DEPRESSED - Stop after each word is printed using PI, PH, PA, PZ, or FP input codes. Depressing START COMPUTE button will cause next word to print.
2	RAISED - 5 stop codes required per line. DEPRESSED - 4 stop codes required per line (no "comments" stop code) using "NO LETTER" code. (SENSE SWITCH 2 must be depressed before transfer to object program.)
4	RAISED - DEPRESSED - Return to start before printing a location.
8	RAISED - Stop before transfer. Depress START COMPUTE button to execute transfer. DEPRESSED - No stop before transfer (input code T).
16	Not used.

OPERATING PROCEDURE

3. SENSE SWITCH OPTIONS (Cont.)

Sense Switch	Function
32	RAISED - Entries in Table of Changes may be output using input code PC or after 16 entries have been made. DEPRESSED - Do not make any entries in Table of Changes when using "NO LETTER" input code. This allows for loading relocatable decimal tapes.

4. PROGRAM STOPS

Error stops are indicated by a number at q=17 (D-sector) of the Command Register. In each instance depressing START COMPUTE button prepares program for input of a new code.

Contents	Meaning
1	Incorrect code word.
2	Machine error.
4	Address error.
5	Stop code missing.
6	Incorrect mnemonic command.
7	Too many address stops set.
8	RS without a corresponding SS.

5. ENTRANCE TO PROGRAM

a. Depress MASTER RESET button on reader.
b. Depress TYPEWRITER TO COMPUTER button on reader.
c. Depress COMPUTER TO TYPEWRITER button on reader.
d. Depress ONE OPERATION button on computer console.
e. Depress SET INPUT MODE button on computer console.
f. Depress EXECUTE LOWER ACCUMULATOR button on computer console.
g. Depress START READ button on reader.

OPERATING PROCEDURE

5. ENTRANCE TO PROGRAM (Cont.)

   h. Manual input light will glow on typewriter. Type in the beginning address in hexadecimal format, and follow with a stop code.
   i. Raise EXECUTE LOWER ACCUMULATOR button on computer console.
   j. Raise ONE OPERATION button on computer console.
   k. Depress START COMPUTE button on computer console.

   "Program Checkout 1" is now ready for an input code.