MDC DEBUGGER 1.0 DOCUMENTATION

This program is prom resident and resides in a 2708 EPROM placed in the fourth prom slot on the MCB. This program is intended as a tool for the debug technician in troubleshooting malfunctioning MDC boards at a system level. It allows the user to read and write specified sectors of the diskette and display the results. It also gives more definitive error messages than the standard 'DISK ERROR' that is usually found.

I. HARDWARE

A known good MCB with the standard 3K monitor proms is required to use this program. A 2708 EPROM or equivalent is used to make the MDC DEBUGGERPROM from the file MDC.DEBUGGER. This prom is placed in the fourth (usually empty) prom socket on the MCB. The debugger software uses a RAM scratch pad area from about 1400 to 1600 hex, so this must contain known good ram.

II. COMMANDS

The program is entered by typing 'J C00' from the debug prompt. The following is a list of the commands available to the user. Only one letter need be entered to execute a command from the program prompt (':').

L -- Loop mode on. This command turns the loop mode on. Once he loop mode is turned on any reads or write will loop until the BREAK button is pressed. Any attempt to exit the loop mode by pressing the reset button could result in destruction of the diskette, if the program is writing to the diskette. When the BREAK is pressed the loop mode is automatically reset. When looping the error messages are turned off until the loop mode is exited.

N -- No loop. Resets loop mode flag.

R -- Read from disk. The user is queried as to the track and sector number to read. The read data is placed in a buffer which can be displayed with the Print command.

W -- Write to disk. The user is queried as to the track and sector number to write to and the hex pattern to write.

P -- Print. Prints the disk buffer. Following a read operation the P command prints the buffer contents to the console. In printing the buffer the current contents is destroyed, so if two successive prints are done the second will always show zeros in the buffer.

A -- Again. Executes the last command that was entered. If the last command was a read or a write then the operation is performed again with the same track and sector arguments.

D -- Debug. Returns to the debug enviroment.

C -- Commands. This displays all of the available commands.

III. ERRORS

The following errors can (and probably will) occur during read and write operations to the disk:

Invalid request This indicates some sort of system problem, such as bad scratch ram or possible a faulty EPROM.

Drive not ready MDC does not see that the drive is ready.

Disk write protected If a write operation is attempted and the MDC thinks the diskette is write protected.

Sector Error The sector address read from the diskette in sector header did not agree with the sector position.

Track Error The track address read from the diskette in sector header did not agree with the head position.

CRC Error A cyclic redundancy check error occurred on the sector read. This indicates one or more data bits in error.

IV. DEBUG STRATGIES

It is recommended that an empty formatted diskette be placed in drive 0 when using this program. Certain sectors should be set up (with a known good system in place) with known patterns (such as AA, 55, 00 and FF hex). Then when a malfunctioning board is put in the system to troubleshoot these sectors can be read and the results compared with what is known to be on the sectors. The loop mode will allow one to get fairly good signals to look at with a scope. By looping on reading a sector that is known to contain a 00 pattern, for example, one will be able to look at many of the read circuits and know immediately if they are working.

GOOD LUCK !!

Dan W. Altman
System Division
Test Engineering