

# Simulator issues

## Simulator Issues

There may be some issues with using the Easy68K as the project's only simulator package that I'm not yet aware of. In general, the simulator is very forgiving when it comes to displaying the disassembled lines of code on its terminal screen. However, what might look OK to you on the screen, may actually appear as one long line of text in the log file because you neglected to add carriage return <CR> and line feed <LF> characters to the end of your output lines. Every line that you send to the screen should be terminated with the ASCII codes for CR and LF.

It is to your advantage to print out the output file. Easy68K has a logging function that will capture everything that was output to the display. If your print-out is messed up, mine will also be messed up.

**There should only be the printable ASCII character set, CR and LF in your I/O to the screen.**

**There will be an automatic deduction of 10 points from your grade if I have to go into the log file and create a printable output file because you forgot to add the proper formatting control characters or you added non-printable characters, such as a beep, to your output.**

## Easy68K bugs

It seems like Easy68k simulator program contains several bugs. Here are the list of Easy68k bugs which are captured by students who have taken this class previously. I strongly urge you to read this first so that you do not have to struggle with this bug when it comes to debugging your own program.

- 1. A MOVE.L operation with direct data going into a data register with the direct data being less than 5 bits causes the EASY68k to perform a MOVEQ instead of a MOVE.L.***
- 2. Similar bugs can be found with most commands which have a "Q" variant. The 68K assembler seems to automatically use the less memory and processor intensive version of the command when possible. I assume this is to maximize the system's efficiency and speed.***

**3. The Easy68k Manual shows that a MOVE operation with the destination being an address register is an invalid operation, and should be a MOVEA operation. However, the Easy68k compiler gives no warning or errors when trying to perform a MOVE with the destination being an address register. Because the Easy68k shows that this kind of operation is invalid, the program will display a MOVEA in place of the MOVE when displaying the data.**

**4. Our inverse assembler will start to act irregularly and will start to display error messages for valid op-code data if a large amount of data/operations are sent through the inverse assembler. We encountered this problem when trying our test routine that tested all possible combinations of valid data for op-codes. We first thought it was our inverse assembler that was causing the irregular performance, but if you repeatedly perform the same action/operation, such as NOP, through the Easy68k assembler for countless times the Easy68k starts to act irregular, which results in our inverse assembler acting irregular as well. The test routine that tested all possible combinations of valid data was around 5,500 lines of code.**