68000 IDE 'C' Compiler and Assembler For use with DE1 Board

Most recent version of the Compiler being tested this term (seems to work)

ide68k30.zip (https://canvas.ubc.ca/courses/130107/files/30213049/download?wrap=1) (https://canvas.ubc.ca/courses/130107/files/30213049/download?download_frd=1) - 68k C compiler V3.3 - Download, unzip and run setup - install to a folder such as "C:\IDE68k" (put a short cut on the task bar for easy access)

Documentation for the Compiler and IDE (User Guide) (https://canvas.ubc.ca/courses/130107/files/30213327/preview)

Using IDE68k to write Assembly Language Programs.docx - (https://canvas.ubc.ca/courses/130107/files/30213554/download?wrap=1) (https://canvas.ubc.ca/courses/130107/files/30213554/download?download_frd=1) Instructions on using the above IDE for Assembly language programming. There is an example 68k assembly language program below to drive the LCD display of a DE1.

Using IDE68k to Write C programs (DE1).docx (https://canvas.ubc.ca/courses/130107/files/30213048/download?wrap=1) (https://canvas.ubc.ca/courses/130107/files/30213048/download?download_frd=1) - Instructions for using the above IDE for writing C programs (you will need the two files below)

When you recompile software in IDE68K and generate a new HEX file for your code, you know from the above file that you can either:

(a) generate a MIF file and recompile your SOF

or

(b) upload the HEX file via Hyperterminal

Both methods take about 5-10 minutes, depending on your code and how fast your computer is.

There is also a third option which is: generate the MIF file and then just update the MIF within the SOF, without doing a full recompile. This takes a much shorter time, under 1 minute. The procedure is as follows:

1. Compile your software via IDE68K, and generate a HEX file

2. Use Visual C++ to convert your HEX file to the MIF file.

3. Go to the start menu and run "Nios II Command Shell (Quartus Prime 18.1)"

4. In the command shell, go to your project directory via the "cd" command (e.g. "cd c:/abc" if that is your project path). Your project directory is where your Quartus ".qpf" file is.

5. Run the command:

quartus_cdb --update_mif <project name>

where <project name> is your QPF file, e.g.:

quartus_cdb --update_mif MC68K.qpf

(if MC68K.qpf is your project name)

6. Run the command:

quartus_asm <project name>

where <project name> is your QPF file, e.g.:

quartus_asm MC68K.qpf

(if MC68K.qpf is your project name)

7. Program the SOF to your card again via the Quartus programmer. The new SOF will include your new software and you should be able to

see that in Hyperterminal, which should automatically reconnect if you have it open already.

C Program Code Files/Examples for use with the IDE68K Compiler

In your Quartus Project (folder Programs->DebugMonitorCode) there should be a **cstart_v4.0 - UserProgram.asm** file that you should use whenever you are creating your own application (i.e. not the debug monitor version of the same file). This one is simpler and assumes Ram in the download area (modify for Sram or Dram based system)

An example C program to flash some lights and write to the LCD display on the 68000/DE1 soft core processor file should also be in your Quartus project:. Look for the file **M68kUserProgram(DE1).c** . Add to an IDE68k project along with the above mentioned **cstart_v4.0 - UserProgram.asm** file