Installing and Using SCASM with Crimson Editor

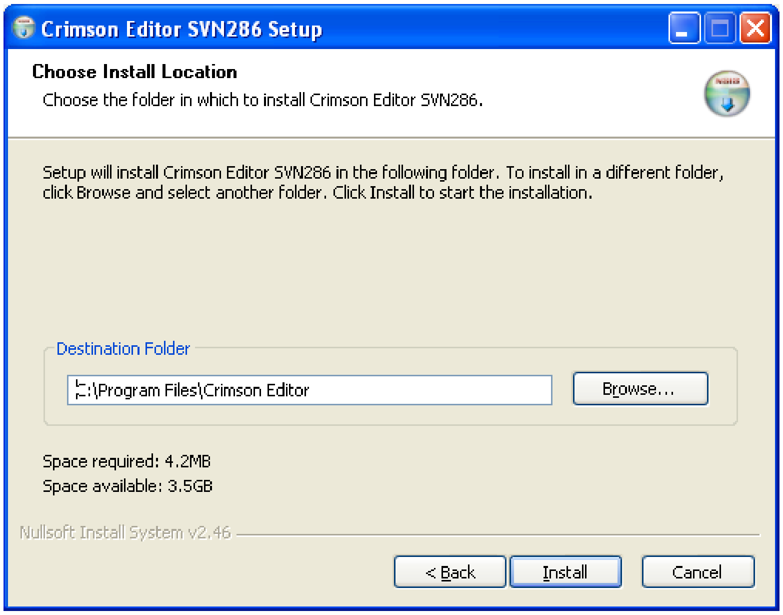
SCASM is a custom assembler that works with the instruction set of the Simple Computer (SCOMP). It saves the user from having to generate machine language instructions manually. Although SCASM is a command-line-driven program, Crimson Editor can be used to provide a color-coded text editor front end, essentially an integrated development environment (IDE) for coding that a user can use alongside the Quartus IDE.

If you are using SCASM and the Crimson Editor in the lab, both programs should already be installed, so skip down to section 2. If you are following the lab instructions to install both programs on your personal computer, continue with section 1.

1. Installing SCASM and Crimson Editor

The first step is to download and install Crimson Editor, which is now maintained as an open-source project at the Emerald Editor page: http://www.emeraldeditor.com/

There is a plan to release an improved product called the Emerald Editor, and it may exist by the time that you read these instructions. But you probably should install the latest Crimson Editor to maximize compatibility, and perhaps the safest way is simply to obtain it from the diglab.gatech.edu class website (under Lab 7 downloads). Currently, the latest version is SVN286, and they have dropped the earlier version designations which ended at 3.72.



Choose Install Location

Choose the folder in which to install Crimson Editor SVN286.

Setup will install Crimson Editor SVN286 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

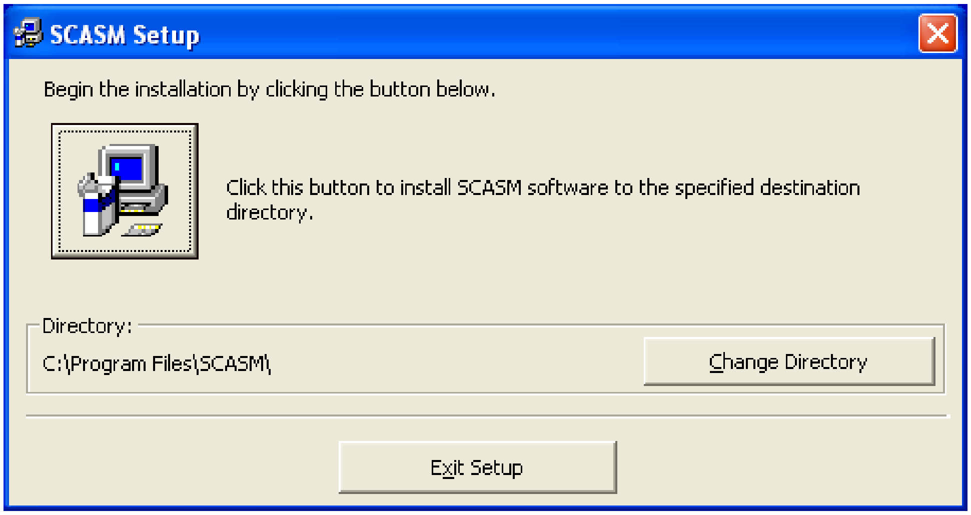
C:\Program Files\Crimson Editor

Space required: 4.2MB
Space available: 3.5GB

Run the downloaded executable. Installation is straightforward, but a good suggestion is to shorten the suggested destination folder to the one shown above.

Next, download the latest version of SCASM from the class website and install it. When selecting an installation directory, it is important to browse to the folder created above as the destination of Crimson Editor, then add the characters \SCASM after it to create a

SCASM subdirectory of the Crimson Editor folder and install there. The default dialog will appear as shown here:

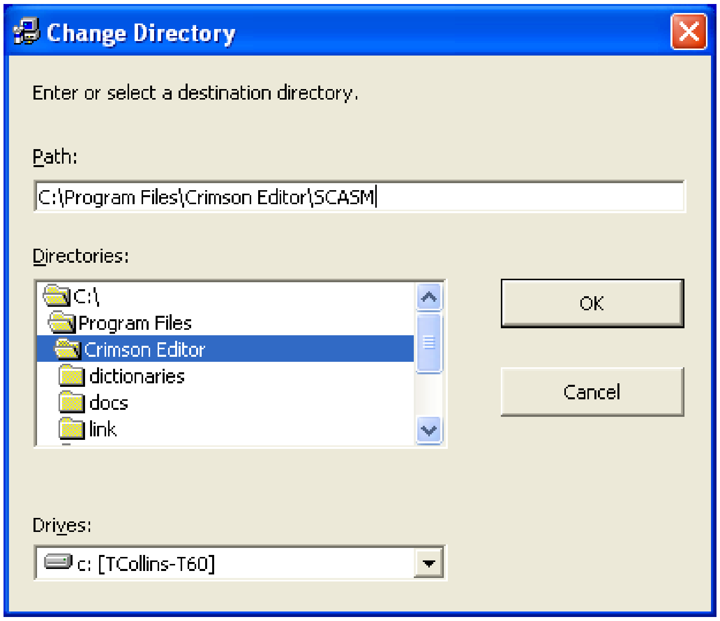

SCASM Setup

Begin the installation by clicking the button below.

Click this button to install SCASM software to the specified destination directory.

Directory: C:\Program Files\SCASM\

Clicking on Change Directory allows you to do this:

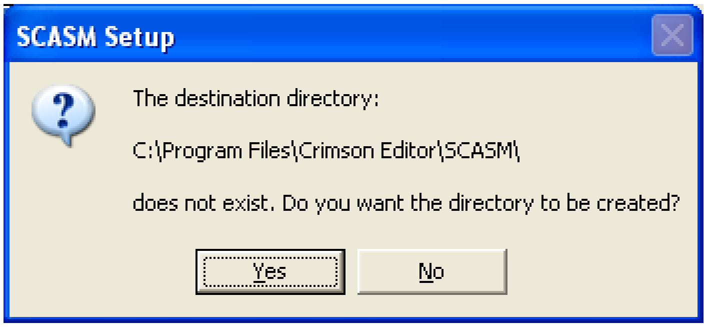

Change Directory

Enter or select a destination directory.

Path: C:\Program Files\Crimson Editor\SCASM

Drives: c: [TCollins-T60]

The installer will ask if you want to create that directory, so allow it (click Yes here):

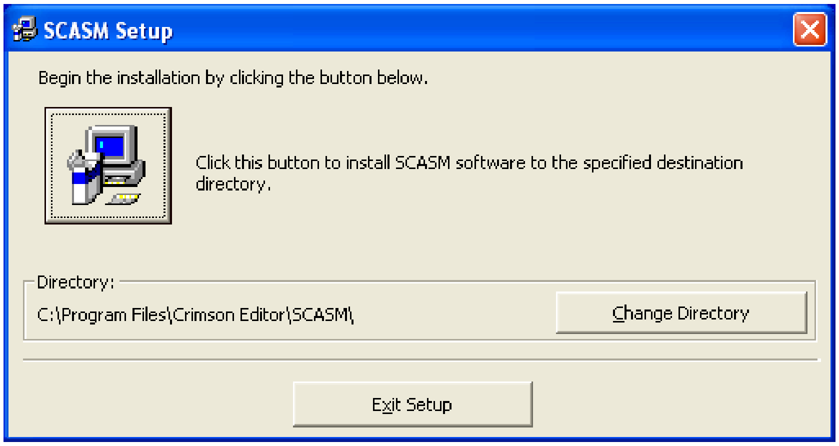

SCASM Setup

The destination directory:

C:\Program Files\Crimson Editor\SCASM\

does not exist. Do you want the directory to be created?

Returning to the setup dialog, the new directory appears:



SCASM Setup

Begin the installation by clicking the button below.

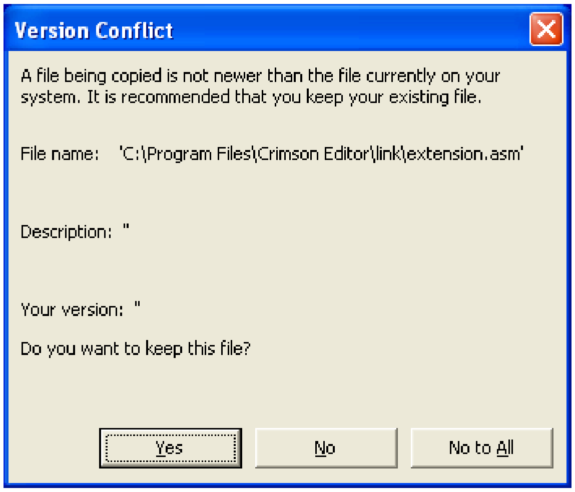
Click this button to install SCASM software to the specified destination directory.

Directory:
C:\Program Files\Crimson Editor\SCASM\

This subdirectory is necessary, because the SCASM installation program will also put Crimson Editor configuration files into several of the Crimson Editor subdirectories, alongside the SCASM directory.

Note that you still have to click the big square button above to begin the installation. This will overwrite the existing EXTENSION.ASM file in the LINK subdirectory, which will also generate a question for you to answer. ALLOW the replacement of EXTENSION.ASM, but you may not want to allow the replacement of DLLs and other files – this is no different than the installation of other programs, and usually newer DLLs are better, so do not replace newer ones with older ones.

Click “No” here to get a new Crimson file.   And click “Yes” here to keep your DLL!



Version Conflict

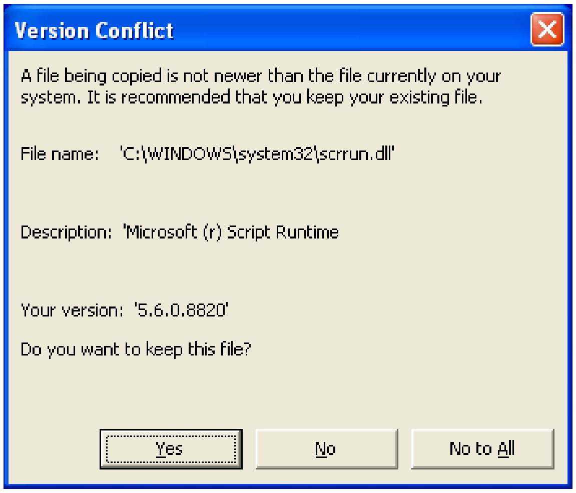
A file being copied is not newer than the file currently on your system. It is recommended that you keep your existing file.

File name: 'C:\Program Files\Crimson Editor\link\extension.asm'

Description: ''

Your version: ''

Do you want to keep this file?



Version Conflict

A file being copied is not newer than the file currently on your system. It is recommended that you keep your existing file.

File name: 'C:\WINDOWS\system32\scrrun.dll'

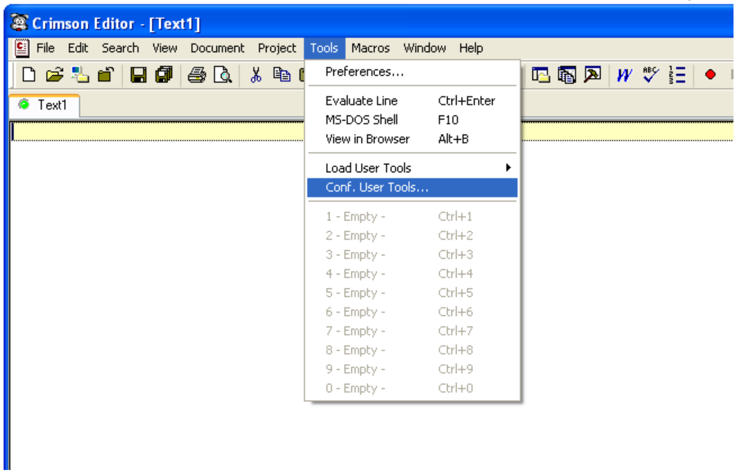
Description: 'Microsoft (r) Script Runtime

Your version: '5.6.0.8820'

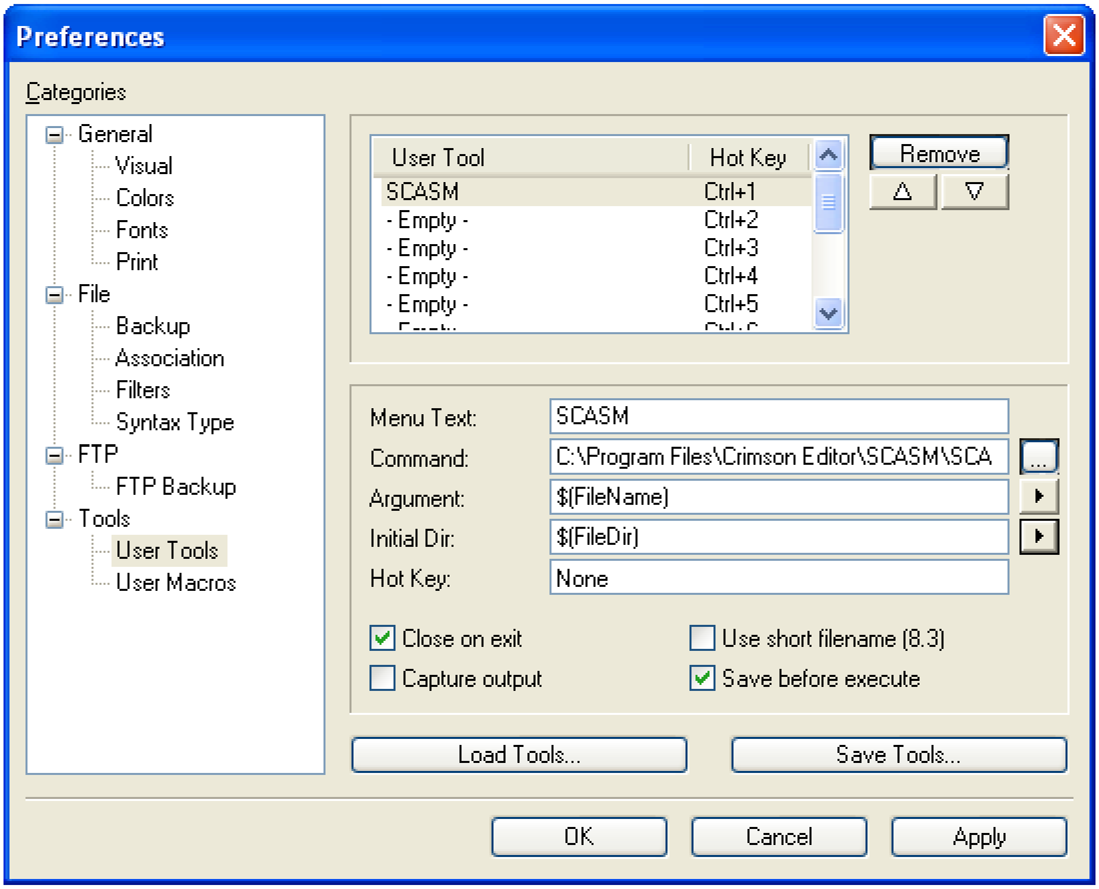
Do you want to keep this file?

1.a Configuring Crimson Editor to invoke SCASM

Start Crimson Editor. You may receive a warning / error message if you have never used Crimson Editor before. Under the **Tools** menu, select **Conf. User Tools...**



With the first empty slot selected, fill in the information as seen below (this assumes Crimson Editor and SCASM were installed to their suggested folder locations). For the Command, enter “C:\Program Files\Crimson Editor\SCASM\SCASM.exe” without the quotes or locate the SCASM.EXE file using the browse button. Press the **OK** button to confirm the changes and save this tool once you have finished. Note the arrow buttons that allow you to pick FileName and FileDir from dropdown menus.



User Tool	Hot Key
SCASM	Ctrl+1
- Empty -	Ctrl+2
- Empty -	Ctrl+3
- Empty -	Ctrl+4
- Empty -	Ctrl+5

Menu Text: SCASM

Command: C:\Program Files\Crimson Editor\SCASM\SCA

Argument: $(FileName)

Initial Dir: $(FileDir)

Hot Key: None

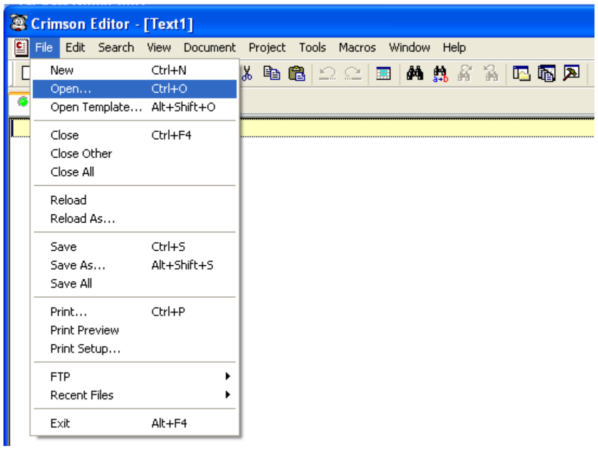
☑ Close on exit ☐ Use short filename (8.3)

☐ Capture output ☑ Save before execute

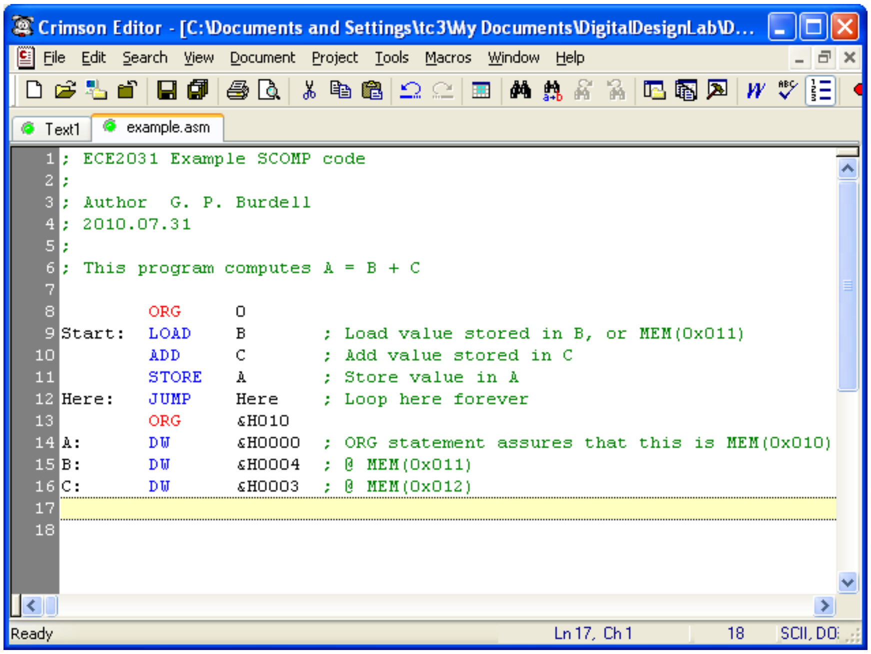
In the steps below, if you do NOT see color-coding that matches the example, then you did not correctly install SCASM in a subdirectory of Crimson Editor, or (if you did) you must not have allowed the installer to copy the new “extension.asm” file which interprets SCASM opcodes and assembler directives. Try uninstalling SCASM and reinstalling it (there is no need to uninstall Crimson Editor).

2. Using SCASM and Crimson Editor

Download and open the example assembly file, EXAMPLE.ASM, from the class website. Open the example file by selecting **Open...** under the **File** menu.



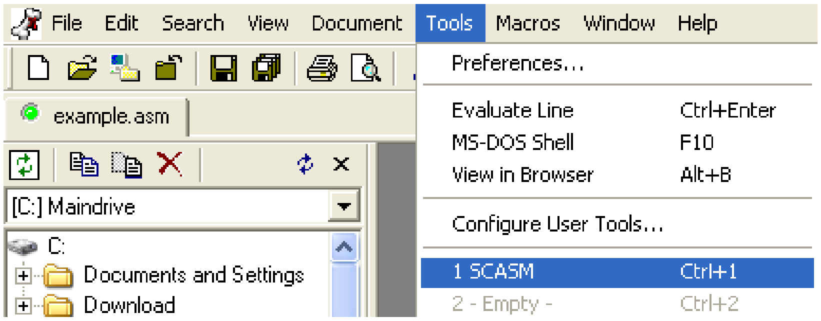
A color-coded assembly file should now be visible in the editor. You may need to turn on line numbers in the View menu to get a similar screen. Green text represents comments. Red text represents assembler directives that do not actually generate code, but tell SCASM what to do. Blue text corresponds to recognized opcodes. Black text represents labels and operands (which may also be labels).


Crimson Editor - [C:\Documents and Settings\tc3\My Documents\DigitalDesignLab\D...
File Edit Search View Document Project Tools Macros Window Help
Text1
example.asm
Ready
Ln 17, Ch 1
18
SCII, DO:

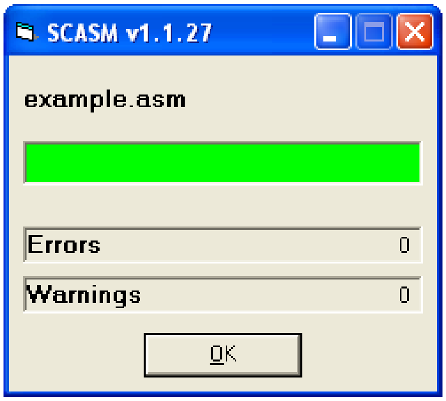
```
; ECE2031 Example SCOMP code
;
; Author  G. P. Burdell
; 2010.07.31
;
; This program computes A = B + C

        ORG     0
Start:  LOAD    B       ; Load value stored in B, or MEM(0x011)
        ADD     C       ; Add value stored in C
        STORE   A       ; Store value in A
Here:   JUMP    Here    ; Loop here forever
        ORG     &H010
A:      DW      &H0000  ; ORG statement assures that this is MEM(0x010)
B:      DW      &H0004  ; @ MEM(0x011)
C:      DW      &H0003  ; @ MEM(0x012)
```

To assemble the file and generate an Altera MIF file, select **SCASM** under the **Tools** menu, or just use the hotkey Ctrl+1, as shown below. If SCASM is not configured as a tool, you may have to follow the steps shown in section 1a above.



Crimson Editor will launch SCASM, which will then assemble the program. Upon completion the SCASM status window should show no errors or warnings.



example.asm

Errors 0

Warnings 0

Had there been one or more errors, SCASM would generate an error file with a .ERR extension, and it would show a RED bar instead of the green bar. Opening the ERR file will provide information, including the line number of the error (Crimson Editor shows the line numbers on the left.) It is often useful to open the MIF files that result from successful compilation, especially if you are getting unexpected results from program execution.