Simple Scalar Installation Tips

Files Needed :

1. simpletools-2v0.tgz from http://www.simplescalar.com/tools.html
2. simplesim-3vd.tgz from http://www.simplescalar.com/tools.html
3. simpleutils-2v0.tgz from http://www.simplescalar.com/tools.html
4. gcc-2.7.2.3.ss.tar.gz from http://american.cs.ucdavis.edu/RAD/gcc-2.7.2.3.ss.tar.gz
5. May need : ar from http://www.ict.kth.se/courses/IS2202/ar
6. May need : ranlib from http://www.ict.kth.se/courses/IS2202/ranlib

LOCAL COPY OF ALL ABOVE FILES

Installation Instructions :

Follow the steps below. Common problems encountered during installation are listed. These should be sufficient to install simplescalar successfully the GCL machines (recent Ubuntu). In case simplescalar throws new errors at you, please mage Google search your friend.

Step 0 :

Find the basics of the system you are working on.
Use following command to check if it's a 64-bit system :
uname -a
If it is a 64-bit system, it might need some changes in the installation procedure.

Unzip all the installation files. Delete the folder gcc-2.6.3

Create the folder where you want to install simplescalar. Move all the uncompressed installation files into it.

Set environment variables as follows:
export IDIR=<your-simplesclar-installation-dir>
export HOST=i686-pc-linux
export TARGET=sslittle-na-sstrix
cd $IDIR

Step 1 :

Ensure required dependencies are installed. You would need *build-essential, flex, bison*.
In some systems, an error of the sort of "cannot find output from lex" may be thrown in spite of flex being installed.
Come around this by creating a symbolic link :
ln -s /usr/local/bin/flex-2.5.4 flex

If doing this after seeing an error in the next step, remove current simpleutils directory & restart Step 3 with a freshly uncompressed copy. In some extreme cases, this would get reflected only after a restart!

Step 2 :

Install SimpleUtils

cd $IDIR/simpleutils-990811

Fix these errors in the source :

- Find *yy_current_buffer* in the file **ld/ldlex.l** & replace it with *YY_CURRENT_BUFFER* . (If this is not done, an error of "`yy_current_buffer' undeclared" would be thrown)

Execute following commands to complete installing :
./configure --host=$HOST --target=$TARGET --with-gnu-as --with-gnu-ld --prefix=$IDIR
make CFLAGS=-O
make install

Step 3 :

Install the simulators

cd $IDIR/simplesim-3.0
make config-pisa
make

This should work without any errors. Test the installation by ;
./sim-safe tests/bin.little/test-math

Step 4 :

Install gcc cross-compiler

cd $IDIR/gcc-2.7.2.3
./configure --host=$HOST --target=$TARGET --with-gnu-as --with-gnu-ld --prefix=$IDIR
chmod -R +w .

Fix these errors in the source :

- In line#60 of file **protoize.c**, replace "*#include <varargs.h>*" with "*#include<stdarg.h>*" (To avoid error claiming GCC no longer implements varargs.h)
- In line#341 of file ~~**decl.c**~~ **obstack.h**, replace "**((void **)__o->next_free)++ = ((void *)datum);*\" with "**((void **)__o->next_free++)=((void *)datum);*\" (To avoid error saying something about "invalid lvalue in increment in function push_class_level_binding"
- Comment out lines 2978-2979 in file **cxxmain.c** (To avoid error that says malloc & realloc have already been declared)
- *cp ./patched/sys/cdefs.h ../sslittle-na-sstrix/include/sys/cdefs.h* (To avoid error looking like "Fix these errors in the source :").

- *cp ../sslittle-na-sstrix/lib/libc.a ../lib/*
- *cp ../sslittle-na-sstrix/lib/crt0.o ../lib/*
- Copy files ar & ranlib into *$IDIR/sslittle-na-sstrix/bin* (To avoid "buffer overflow" error involving ar or ranlib)

[The last three may not be needed in some systems. But are needed in the latest Ubuntu systems].

Now run :
make LANGUAGES=c CFLAGS=-O CC="gcc -m32"

This should throw an error : "insn-output.c:xyz: missing terminating " character".
Insert \ at the end of lines 675, 750 & 823 of file **insn-output.c**. The lines should look like *"return "FIXME\n\"*

Now run :
make LANGUAGES=c CFLAGS=-O CC="gcc -m32"
make enquire
$IDIR/simplesim-3.0/sim-safe ./enquire -f > float.h-cross
make LANGUAGES=c CFLAGS=-O CC="gcc -m32" install

In the GCL machines, that should see you through. Other problems that you are likely to encounter are below :

- If it's a 64-bit OS, you might need to use *CC="gcc -m64"* or *CC="gcc"* in the above make commands. Otherwise, it complains that "stubs-64.h" isn't found.
- If the error looks like "<path>/stdio.h:167: parse error before `void'", try editing **Makefile** at line#130 & append *-I/usr/include* to the end of that line.
- For error about "libgcc2.c:484: `BITS_PER_UNIT' undeclared", change line#97 of file **libgcc2.c** to *"#define BITS_PER_UNIT 8"*
- For "sendmsg.c:36: parse error", in file **objc/sendmsg.c**, insert "*#define INVISIBLE_STRUCT_RETURN 0*" at line 35, and delete lines 36 to 40 (inclusive).

Step 5 :

Try it out. Simplescalar is installed now.

In a new directory, create a simple hello world C program - hello.c
Compile it using the cross-compiler with the following command :
$IDIR/bin/sslittle-na-sstrix-gcc -o hello hello.c

Run it using a simulator :
$IDIR/simplesim-3.0/sim-safe hello

You should see some statistics after the lines "sim: **starting functional simulation** Hello World!" as the output.

Congrats!

If you want to really relax, try adding the simplescalar simulator & cross-compiler directories to your PATH so as to avoid typing the entire path each time :
export PATH=$PATH:$IDIR/bin/:$IDIR/simplesim-3.0

Add this line to your ~/.bashrc file