

Simplescalar installation instructions

Mats Brorsson, Feb 24, 2009

Introduction

The following text describes the procedure of installing the simulator Smplescalar [1] on a modern Linux distribution. The procedure has been tested on Ubuntu Linux 8.04 (Hardy Heron) but should work on most Linux distributions. If you use Ubuntu and 8.10 (Intrepid Ibex) or later, you may experience some new problems for which there is a solution as described in the footnote on page 4. The instructions are modified from [2].

Preparations

Download the necessary source code files:

- **simpletools-2v0.tgz** from <http://www.simplescalar.com/tools.html>
- **simplesim-3v0d-with-cheetah.tar.gz** from <http://www.ict.kth.se/courses/IS2202/software>
- **simpleutils-990811.tar.gz** from <http://www.eecs.umich.edu/mirv/>
- **gcc-2.7.2.3.ss.tar.gz** from <http://american.cs.ucdavis.edu/RAD/gcc-2.7.2.3.ss.tar.gz>

Setup some environment variables (assuming your home directory is /home/matsbror and that you are using a bash-compatible shell¹):

```
$ export IDIR=/home/matsbror/simplescalar
$ export HOST=i686-pc-linux
$ export TARGET=sslittle-na-sstrix
```

Create the directory “**simplescalar**” under your home directory, and copy all the four tarballs into it:

```
$ mkdir $IDIR
$ mv simplesim-3v0d-with-cheetah.tgz $IDIR
$ mv simpletools-2v0.tgz $IDIR
$ mv simpleutils-990811.tar.gz $IDIR
$ mv gcc-2.7.2.3.ss.tar.gz $IDIR
```

¹ If you use tcsh or the like, the only difference should be in how environment variables are set.

Update Ubuntu (or whatever distribution you are using) with the following packages:

- build-essential
- flex
- bison

You can use the command “`sudo apt-get install <package name>`” to retrieve and install these packages.

Want to run on a Windows computer?

If you don't have access to a Linux computer and you want to use a Windows computer instead, you can use a virtual machine. A model which has been tested to work is to use VmWare-server which can be downloaded for free at <http://www.vmware.com>.

Installing SimpleTools

Just un-pack the package file, and remove the old gcc folder:

```
$ cd $IDIR
$ tar xvfz simpletools-2v0.tgz
$ rm -rf gcc-2.6.3
```

Installing SimpleUtils

First un-pack the package file.

```
$ cd $IDIR
$ tar xvfz simpleutils-990811.tar.gz
$ cd simpleutils-990811
```

Before building the code, you need to fix some sources of errors:

- In directory `ld` find file `ldlex.l` and replace all instances of `yy_current_buffer` with `YY_CURRENT_BUFFER`.

```
$ ./configure --host=$HOST --target=$TARGET --with-gnu-as
--with-gnu-ld --prefix=$IDIR
$ make CFLAGS=-O
$ make install
```

If you experience compilation problems with “`'yy_current_buffer ' undeclared`” there may be problems with flex. Download and install the following version which is known to work

together with Simplescalar. <http://www.ict.kth.se/courses/IS2202/flex.tar.gz>. The following command(s) should work for installation: “./configure; make; sudo make install”.

After installation of flex, if needed you should remove the installation directory of simpleutils and restart unpacking, configuration etc.

If you have a multicore processor, you may speed up the compilation process by using “make -j 2” for a dual core and “make -j 4” for a quad core. It may even go faster if you specify a number higher than the core count as compilations are I/O intensive and you then may overlap CPU-intensive parts of the build process with disk operations.

Building the simulators

Un-pack the simulator package.

```
$ cd $IDIR
$ tar xvfz simplesim-3v0d-with-cheetah.tar.gz
$ cd simplesim-3.0
$ make config-pisa
$ make
```

You may test the installation of simplesim by:

```
$ ./sim-safe tests/bin.little/test-math
```

Installation of gcc cross-compiler

Un-pack the source code and configure the installation:

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

Now, you need to modify a number of files in order for the compilation process to work.

- Change the **Makefile** at line 130, by appending `-I/usr/include` to the end of the line
- Edit line 60 of **protoize.c**, and replace “`#include <varargs.h>`” with “`#include <stdarg.h>`”

- To fix an error message about **decl.c**, saying “invalid lvalue in increment”, do the following: Edit **obstack.h** at line 341 and change:
 - `*((void **)__o->next_free)++=((void *)datum);\`
 - with
 - `*((void **)__o->next_free++)=((void *)datum);\`

Also run the following commands:

- `cp ./patched/sys/cdefs.h ../sslittle-na-sstrix/include/sys/cdefs.h`
- `cp ../sslittle-na-sstrix/lib/libc.a ../lib/`
- `cp ../sslittle-na-sstrix/lib/crt0.o ../lib/`

Next build the compiler:

```
$ make LANGUAGES=c CFLAGS=-O CC="gcc -m32"
```

This command will lead to a number of error messages which requires you to edit a couple of files:

- append ‘\’ at the end of lines 675, 750 and 823 in file **insn-output.c**²
 - o Do “make LANGUAGES...” again and then
- remove lines 2978-2979 in file **cxmain.c**, if needed

Now, run:

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

That should execute with no troubles.

2 At this point you might encounter a “buffer overflow” if you use Ubuntu 8.10 or later. If so, download the following files and put them in \$IDIR/sslittle-na-sstrix/bin:
 - <http://www.ict.kth.se/courses/IS2202/ar>
 - <http://www.ict.kth.se/courses/IS2202/ranlib>

With that, Simple Scalar has been installed, along with its tools, utils, and compilers. Let's now test compiling a new code and running it on top of simplescalar:

In a separate directory, create a new file, **hello.c**, that has the following code:

```
#include<stdio.h>
main()
{
    printf("Hello World!\n");
}
```

then compile it using the following command:

```
$ $IDIR/bin/sslittle-na-sstrix-gcc -o hello hello.c
```

That should generate a file **hello**, which we will run over the simulator:

```
$ $IDIR/simplesim-3.0/sim-safe hello
```

In the output, you should be able to find the following:

```
sim: ** starting functional simulation **
Hello World!
```

Your work is done!

References

- [1] **SimpleScalar**. SimpleSclar home page. [Online] [Cited: March 26, 2008.] <http://www.simplescalar.com>.
- [2] **Al-Issa, Khalid**. A Simplescalar Installation guide. [Online] [Cited: den 26 March 2008.] <http://www.studiokhalid.com/simplescalar/simplescalar.htm>.