1. Installation issues
    1. OS version misdetection
    2. Python version
    3. Utilities compilation going wrong
    4. BFD using symbols from libiberty
2. Cross-compilation issues
    1. Wrong cross-compiler
3. Simulation issues
    1. Bus simulation and multiple drivers
    2. Invalid ELF format

Installation issues

OS version misdetection

Linux 64 bits is misdetected most of the time, in SystemC configuration section in `soclib/utils/conf/soclib.conf`, change

```
os = _platform(),
```

to

```
os = 'linux64',
```

Python version

```
Traceback (most recent call last):
  File
"/home/share/dcis_scme/CI/FAUST/PROJETS/SOCLIB/soclib/utils/lib/python/soclib-cc-main.py",
line 32, in ?
    from soclib_cc.builder.cxx import CxxCompile
  File
"/home/share/dcis_scme/CI/FAUST/PROJETS/SOCLIB/soclib/utils/lib/python/soclib_cc/builder/cxx.py"
line 29, in ?
    import action
  File
"/home/share/dcis_scme/CI/FAUST/PROJETS/SOCLIB/soclib/utils/lib/python/soclib_cc/builder/action.
line 79
    @classmethod
    ^
SyntaxError: invalid syntax
```

Please update to python-2.4

Utilities compilation going wrong

When I try to compile the Soclib tools I am getting the following error:

```
make[1]: Entering directory `/lip6/soclib/utils/src/pipe2fb'
make[1]: Nothing to be done for `Makefile'.
make[1]: Leaving directory `/lip6/soclib/utils/src/pipe2fb'
make[1]: Entering directory `/lip6/soclib/utils/src/xtty'
gcc -c -o xtty.o xtty.c  -Wall -O2
gcc -o soclib-xtty xtty.o -L/usr/X11R6/lib -lX11 -lXpm
make[1]: Leaving directory `/lip6/soclib/utils/src/xtty'
```

```
make[1]: Entering directory `/lip6/soclib/utils/src/fb_screen'
cc `sdl-config --cflags` -Wall -O2  linux -c -o fb.o fb.c
gcc: linux: No such file or directory
make[1]: *** [fb.o] Error 1
make[1]: Leaving directory `/lip6/soclib/utils/src/fb_screen'
make: *** [all] Error 1
```

It looks like you have the `TARGET_ARCH` environment variable set. Make is doing some silly replacements when it is set. Unset it and try again:

```
$ unset TARGET_ARCH
$ make
```

BFD using symbols from libiberty

Lots of errors like

```
/usr/lib/gcc/x86_64-redhat-linux/3.4.6/../../../../lib64/libbfd.a(archures.o)(.text+0x3e0): In f
: undefined reference to `_sch_istable'
```

can be solved with

```
config.default.toolchain.libs.append('-liberty')
```

at the end of `soclib/utils/conf/soclib.conf`

Cross-compilation issues

Wrong cross-compiler

In a bunch of GCC error messages, you see

```
as: unrecognized option `-EL'
```

Your mips gcc has been created wrongly, it's using your native (most probably x86) assembler.

This is most probably because you configured `binutils` and `gcc` with different `--prefix=` options.

Recreate your cross-compilation toolchain.

Simulation issues

Bus simulation and multiple drivers

Using PiBus under SystemC-2.2, you see

```
Error: (E115) sc_signal<T> cannot have more than one driver
 signal `pibus_d' (sc_signal)
 first driver `tty_wrapper.port_8' (sc_inout)
 second driver `multiram_wrapper.port_8' (sc_inout)
 In file: ../../../../src/sysc/communication/sc_signal.cpp:137
```

SystemC got picky about multiple drivers. This probably needs a fix in simulation models, using SystemC-2.1 is a

workaround for now.

Invalid ELF format

You get the following exception:

```
terminate called after throwing an instance of 'soclib::exception::RunTimeError'
 what():  BFD failed: File in wrong format
Abort trap
```

You probably linked you simulator with native libbfd (your host's one). Try to configure soclib-cc to link against you cross-compilation toolchain libbfd.