

Platform Compilation

Prerequisites

We'll assume your configuration file is valid.

In order to compile a platform with soclib-cc, you need:

- Your *top-cell* SystemC netlist,
- A platform description file, listing all used components and their parameters.

Your *top-cell* file should

- Include all soclib component definitions
 - ◆ Paths are relative to systemc/include
- Define a `sc_main`,
- Create the components, signals, ...
- Connect them
- Run simulation.

Your platform description should be a python-parseable file, defining variable `todo`, instance of type `Platform`. Arguments passed to `Platform` are the used components, and you custom files (You may want to link in more than one custom `.cc` file).

Example

I will only use a mapping table, instantiate it, and compile a platform:

File `top.cc`:

```
#include <iostream>
#include <systemc.h>
#include "mapping_table.h"

int sc_main(int argc, char **argv)
{
    using soclib::common::IntTab;
    using soclib::common::MappingTable;
    using soclib::common::Segment;

    MappingTable mt(32, IntTab(2), IntTab(2), 0x300000);
    mt.add( Segment("reset", 0xbfc00000, 0x800, IntTab(0), true) );
    mt.add( Segment("excep", 0x80000080, 0x100, IntTab(0), true) );
    mt.add( Segment("data", 0x40000000, 0x4000, IntTab(1), true) );
    mt.add( Segment("code", 0x04000000, 0x4000, IntTab(2), true) );

    std::cout
        << "Mapping table is:" << std::endl
        << mt << std::endl;
    return 0;
}
```

File `platform_desc`:

```
todo = Platform('caba', 'top.cc',
               uses = [
```

```

        Uses('common:mapping_table'),
    ])

```

Now the compilation:

```

$ soclib-cc -p platform_desc
soclib-cc: Entering directory `/Users/nipo/labo/soclib/rose/soclib'
CxxSource gen repos/release/soclib::common::MappingTable_mapping_table.cc
CreateDir gen repos/release
CxxSource gen repos/release/soclib::common::AddressDecodingTable_input_t=unsigned_int,output_t=b
CxxSource gen repos/release/soclib::common::AddressDecodingTable_input_t=unsigned_int,output_t=i
CxxSource gen repos/release/soclib::common::AddressMaskingTable_data_t=unsigned_int_address_mask
CxxSource gen repos/release/soclib::common::Segment_segment.cc
CxxCompile compile repos/release/soclib::common::MappingTable_mapping_table.o
CxxCompile compile repos/release/soclib::common::AddressDecodingTable_input_t=unsigned_int,output
CxxCompile compile repos/release/soclib::common::AddressDecodingTable_input_t=unsigned_int,output
CxxCompile compile repos/release/soclib::common::AddressMaskingTable_data_t=unsigned_int_address
CxxCompile compile repos/release/soclib::common::Segment_segment.o
CxxCompile compile top.o
CxxLink link system.x
$

```

And running:

```

$ ./system.x

SystemC 2.2.05jun06_beta --- Aug 29 2006 16:42:34
Copyright (c) 1996-2006 by all Contributors
ALL RIGHTS RESERVED

Mapping table is:
Mapping table: ad:(2)
<Segment "reset": @0xbfc00000, 0x800 bytes, (0), cached>
<Segment "excep": @0x80000080, 0x100 bytes, (0), cached>
<Segment "data": @0x40000000, 0x4000 bytes, (1), cached>
<Segment "code": @0x4000000, 0x4000 bytes, (2), cached>

$

```

For complete examples, watch in `/path/to/soclib/platforms/`.