

SystemC - Hierarchies (05A)

SystemC

Copyright (c) 2012 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Based on the following original work

- [1] Aleksandar Milenkovic, 2002
CPE 626 The SystemC Language – VHDL, Verilog Designer's Guide
<http://www.ece.uah.edu/~milenska/ce626-02S/lectures/cpe626-SystemC-L2.ppt>
- [2] Alexander de Graaf, EEMCS/ME/CAS, 2010
SystemC: an overview ET 4351
ens.ewi.tudelft.nl/Education/courses/et4351/SystemC-2010v1.pdf
- [3] Joachim Gerlach, 2001
System-on-Chip Design with System of Computer Engineering
<http://www2.cs.uni-paderborn.de/cs/ag-hardt/Forschung/Data/SystemC-Tutorial.pdf>
- [4] Martino Ruggiero, 2008
SystemC
polimage.polito.it/~lavagno/codes/SystemC_Lezione.pdf
- [5] Deepak Kumar Tal, 1998-2012
SystemC Tutorial
<http://www.asic-world.com/systemc/index.html>

Module Hierarchy

Modules may contain sub-modules (hierarchical structure)

In **SC_MODULE**:

```
// sub-module declaration  
module_type *my_module;
```

In **SC_CTOR** of **SC_MODULE**:

```
// sub-module instantiation and port mapping  
SC_CTOR( module_name ) {  
    my_module = new module_type ( "label");  
  
    my_module -> in1 (sig1);  
    my_module -> in2 (sig2);  
    my_module -> out1 (sig3);  
}
```

Module Instantiation

Direct Instantiation

```
SC_MODULE(ex2)
{
    ex1 ex1_instance;

    SC_CTOR(ex2) : ex1_instance("ex1_anyname")
    {
        // body
    }
}
```

Indirect Instantiation

```
SC_MODULE(ex2)
{
    ex1 *ex1_instance;

    SC_CTOR(ex2)
    {
        ex1_instance = new("ex1_anyname");
        // body
    }
}
```

Module Signals

Signals are used to connect ports of lower-level modules together

- local to a module
- carry data (ports: the direction of data)
- connected to ports in 2 ways:
 - Positional connection
 - Named connection

Positional Connection

```
// filter.h
#include "systemc.h"
#include "mult.h"
#include "coeff.h"
#include "sample.h"
```

```
SC_MODULE(filter) {
    sample *s1;
    coeff  *c1;
    Mult   *m1;

    sc_signal<sc_uint<32>> q, s, c;

    SC_CTOR(filter) {
        s1 = new sample ("s1");
        (*s1)(q,s);

        c1 = new coeff ("c1");
        (*c1)(c);

        m1 = new mult ("m1");
        (*m1)(s,c,q);
    }
}
```

Named Connection

```
// filter.h
#include "systemc.h"
#include "mult.h"
#include "coeff.h"
#include "sample.h"
```

```
SC_MODULE(filter) {
    sample *s1;
    coeff *c1;
    Mult *m1;

    sc_signal<sc_uint<32>> q, s, c;

    SC_CTOR(filter) {
        s1 = new sample ("s1");
        s1->din(q);
        s1->dout(s);

        c1 = new coeff ("c1");
        c1->out(c);

        m1 = new mult ("m1");
        m1->a(s);
        m1->b(c);
        m1->q(q);
    }
}
```


References

- [1] Aleksandar Milenkovic, 2002
CPE 626 The SystemC Language – VHDL, Verilog Designer’s Guide
<http://www.ece.uah.edu/~milenska/ce626-02S/lectures/cpe626-SystemC-L2.ppt>

- [2] Alexander de Graaf, EEMCS/ME/CAS, 2010
SystemC: an overview ET 4351
ens.ewi.tudelft.nl/Education/courses/et4351/SystemC-2010v1.pdf

- [3] Joachim Gerlach, 2001
System-on-Chip Design with System of Computer Engineering
<http://www2.cs.uni-paderborn.de/cs/ag-hardt/Forschung/Data/SystemC-Tutorial.pdf>

- [4] Martino Ruggiero, 2008
SystemC
polimage.polito.it/~lavagno/codes/SystemC_Lezione.pdf

- [5] Deepak Kumar Tal, 1998-2012
SystemC Tutorial
<http://www.asic-world.com/systemc/index.html>