



- Ch1 Overview of System Design Using SystemC
- Ch2 Overview of SystemC
- Ch3 Data Types
- Ch4 Modules
- Ch5 Notion of Time
- Ch6 Concurrency
- Ch7 Predefined Channels



- Ch8 Structure
- Ch9 Communication
- Ch10 Custom Channels and Data
- Ch11 Transaction Level Modeling

Copyright © F. Muller 2005-2020





1





Predefined Primitive Channels (Mutexs, FIFOs, Signals)				
Simulation Kernel	Threads & Methods	Channels & Interfaces	Data types Logic, Integers, Fixed point	
	Events, Sensitivity & Notification	Modules & Hierarchy		

Copyright © F. Muller 2005-2020

(SYSTEM C™ Ch7 - 2 -

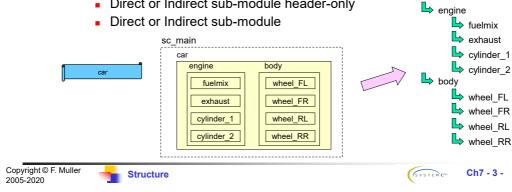


Module Hierarchy



sc_main

- Systems require partitioning and hierarchy
 - complexicity
 - better understanding
 - project management
- Implementations
 - Direct or Indirect top-level
 - Direct or Indirect sub-module header-only



3

Direct or Indirect top-level



Direct top module

```
int sc_main(int argc, char* argv[])
                                                             sub-design instances are instantiated
                                                             and initialized in one statement
     Wheel Wheel_FL("Wheel_FL"); Wheel Wheel_FR("Wheel_FR");
     sc_start();
```

Indirect top module

```
int sc_main(int argc, char* argv[])
                                                              pointer declaration
                            Wheel *wheel_FL;
                            Wheel *wheel_FR;
                                                                                  instance creation
                            wheel_FL = new Wheel("Wheel_FL");
                            wheel_FR = new Wheel("Wheel_FR");
                            sc_start();
                            delete wheel FL;
                            delete wheel_FR;
                            return 0;
Copyright © F. Muller
                                                                                                       (SYSTEM C™ Ch7 - 4 -
                               Structure
```





Direct or Indirect sub-module header-only

```
Direct sub-module header-only
                                                                          Indirect sub-module header-only
                     Body.h
                                                                                Body.h
               SC_MODULE(Body)
                                                                           SC_MODULE(Body)
                    // Sub-module instances
                                                                               // Sub-module instances
                                                                                                              pointer declaration
                    Wheel Wheel_FR;
                                                   declaration
                                                                                Wheel *wheel_FR;
                    Wheel Wheel_FL;
Wheel Wheel_RR;
                                                                               Wheel *wheel_FL;
Wheel *wheel_RR;
                    Wheel Wheel_RL;
                                                                               Wheel *wheel_RL;
                    // Constructor
                                                                                                               instance creation
                                                                               // Constructor
                    SC_CTOR(Body)
                                                                               SC CTOR(Body)
                                                     initialization
                        : Wheel_FL("Wheel_FL"),
Wheel FR("Wheel FR"),
                                                                                    wheel FL = new Wheel("Wheel FL");
                        Wheel_RL("Wheel_RL")
                                                                                    wheel FR = new Wheel ("Wheel FR");
                        Wheel_RR("Wheel_RR")
                                                                                     wheel_RL = new Wheel("Wheel_RL");
                                                                                    wheel_RR = new Wheel("Wheel_RR");
              };
                                                                               // Destructor
                                                                               ~Body()
                                                                               {
                                                                                     delete wheel_FR;
                                                                               }
Copyright © F. Muller
2005-2020
                                                                                                        (SYSTEM C™ Ch7 - 5 -
                               Structure
```



5

Direct or Indirect sub-module With Separate Compilation



Direct sub-module

Indirect sub-module

```
Body.h
                                                                                     Body.h
               SC MODULE(Body)
                                                                               SC_MODULE(Body)
                     // Sub-module instances
                                                                                    // Sub-module instances
                                                      declaration only
                     Wheel Wheel FR;
                                                                                                                      pointer declaration
                                                                                    Wheel *wheel_FR;
Wheel *wheel_FL;
                     Wheel Wheel_FL;
                     Wheel Wheel RR:
                                                                                    Wheel *wheel_RR;
                     Wheel Wheel_RL;
                                                                                    Wheel *wheel_RL;
                    // Constructor
SC_HAS_PROCESS(Body);
                                                                                    SC_HAS_PROCESS(Body);
                                                                                                                      instance creation
                     Body(sc_module_name nm);
                                                                                    Body(sc_module_name nm);
                                                  best approaches for IP!
                                                                                     Body.cpp
                      Bodv.cpp
                                                  pre-compiled object files // Constructor
               // Constructor
                                                                               Body::Body(sc_module_name nm)
               Body::Body(sc_module_name nm)
                             : Wheel_FL("Wheel_FL"),
Wheel_FR("Wheel_FR"),
Wheel_RL("Wheel_RL"),
                                                                                             : sc_module(nm)
                                                                               {
                                                                                             wheel_FL = new Wheel("Wheel_FL");
                                                                                            wheel_FR = new Wheel("Wheel_FR");
wheel RL = new Wheel("Wheel RL");
                             Wheel_RR("Wheel_RR")
               sc module(nm)
                                                                                             wheel_RR = new Wheel("Wheel_RR");
Copyright © F. Muller
                                                                                                               (SYSTEM C™ Ch7 - 6 -
                                 Structure
```

6





Level	Allocation	+	-
Main	Direct	Least code	Inconsistent with other levels
Main	Indirect	Dynamically configurable	Involves pointers
Module	Direct header-only	- All in one file - Easier to understand	Requires sub-module headers
Module	Indirect header-only	- All in one file - Dynamically configurable	Involves pointers
Module	Direct sub-module (separate compilation)	Hides implementation	Requires sub-module headers
Module	Indirect sub-module (separate compilation)	- Hides sub-module headers and implementation - Dynamically configurable	Involves pointers



(SYSTEM C™ Ch7 - 7 -