INSTRUCTIONS FOR SSGEN ENHANCEMENT

MAR. 30, 2018

DESIGN METHODOLOGY DEPARTMENT
RENESAS ELECTRONICS CORPORATION

INSTRUCTIONS

- The followings should be done.
  - To improve feature of verification environments generation
  - To update SSGEN user guide

- Points to keep in mind
  - When you cannot handle the request or problem from HLD designers, please ask REL for support
  - When you ask REL for help, please explain what you have considered, and distinct the points you have resolved and that you could not resolve

CONTENTS OF THIS ACTIVITY

1. Improve feature of verification environments generation
2. Other enhancements
3. SSGEN implement
4. SSGEN test
5. Update SSGEN user guide

CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**

- Prepare a new command to change main_*.cpp generation way
  - Purpose: to keep user modification of sc_main function
- Support “#include” and “cthread” in `ifdef TESTBENCH
  - Purpose: to make user modify TB easily
- Build cocotb-SystemC environment
  - Purpose: to introduce cocotb (HDL verification framework) for Unit Test quality improvement

CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**

- Prepare a new command to change main_*.cpp generation way (1/3)
  - Add new command **separate_scmain_file {on|off}**
    - **off** : default
      - ✓ Output main_*.cpp only as before
    - **on** : Output sc_main descriptions into two files (main_*.hpp and main_*.cpp)
      - ✓ main_*.hpp
        - ✓ Output only following descriptions
          - Instantiations of DUT, TB and memory model
          - sc_signal declarations
          - Signal connections
        - ✓ File overwrite rule is same as *.h file (rename existed main_*.hpp to main_*.hpp.*N*)

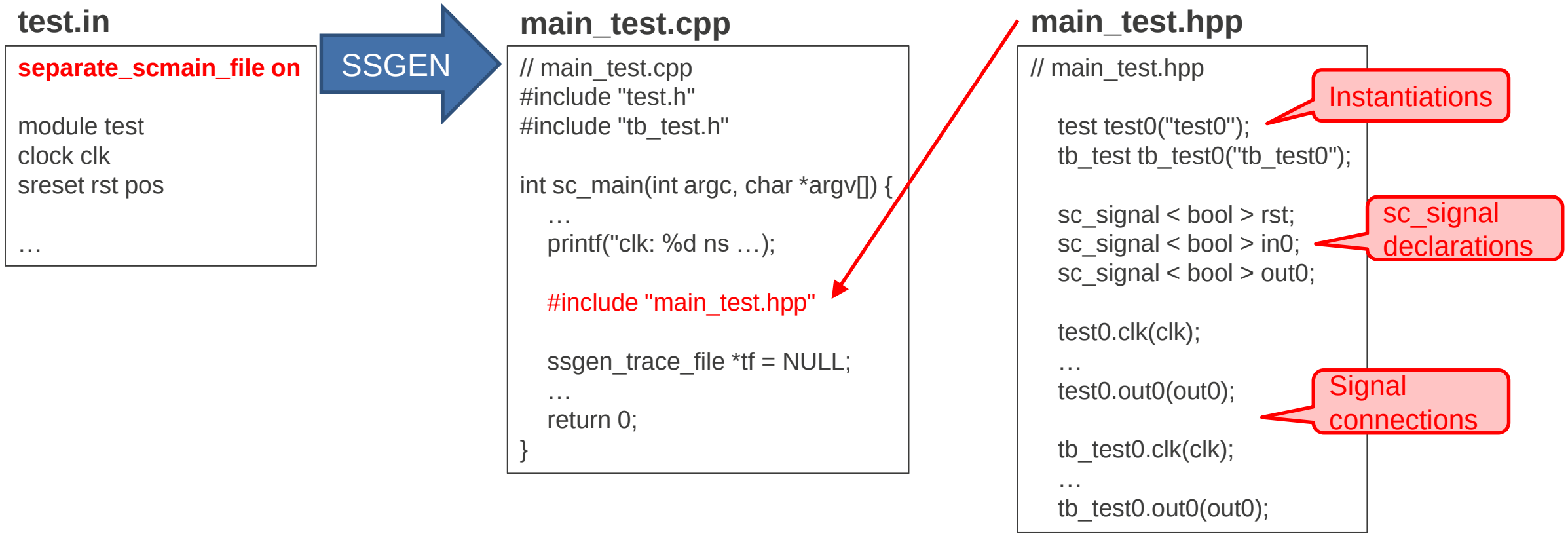
CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**

- Prepare a new command to change main_*.cpp generation way (2/3)
  - Add new command **separate_scmain_file {on|off}**
    - **on** : Output sc_main descriptions to two files (main_*.hpp and main_*.cpp) (Cont.)
      - ✓ main_*.cpp
        - ✓ Output other descriptions
        - ✓ Output #include main_*.hpp
        - ✓ File overwrite rule is same as *.cpp file (Generate main_*.cpp_tmp if main_*.cpp already exists)

CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**
   - Prepare a new command to change main_*.cpp generation way (3/E)

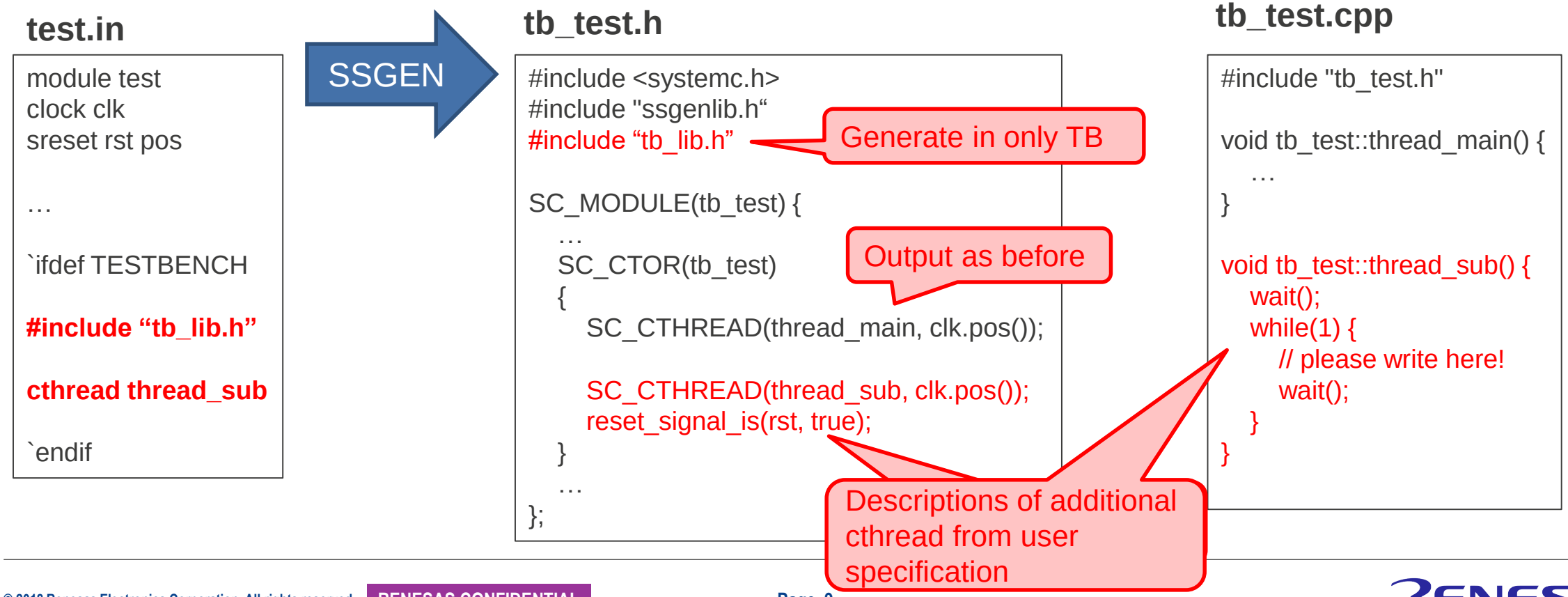


CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**
   - Support “#include” and “cthread” in `ifdef TESTBENCH (1/2)
     - **#include**
       - When “#include” is specified inside of `ifdef TESTBENCH, the “#include” description is generated in TB only.
     - **cthread**
       - When “cthread” is specified inside of `ifdef TESTBENCH, SC_CTHREAD descriptions are generated in TB in addition to “thread_main”

CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**
   - Support “#include” and “cthread” in `ifdef TESTBENCH (2/E)



CONTENTS OF THIS ACTIVITY

1. Improve feature of verification environments generation

- Build cocotb-SystemC environment (1/6)
  - cocotb: Coroutine Co-simulation Test Bench

    https://github.com/potentialventures/cocotb

    http://cocotb.readthedocs.io/en/latest/introduction.html#what-is-cocotb

    ✓ Cocotb is a COroutine based COsimulation TestBench environment for verifying VHDL/Verilog RTL using Python.

    ✓ Cocotb is completely free, open source (under the BSD License) and hosted on GitHub.

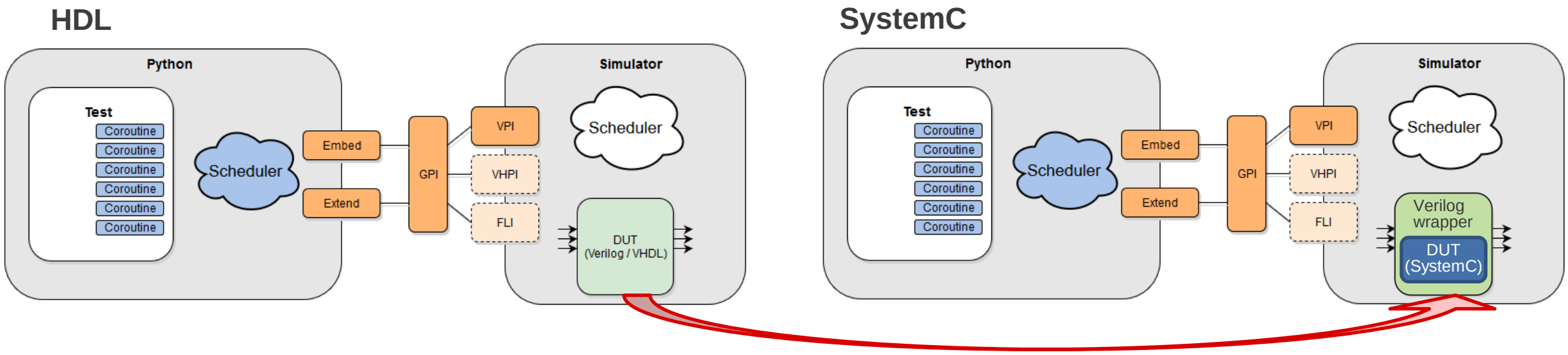
    ✓ Cocotb requires a simulator to simulate the RTL.

CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**
   - Build cocotb-SystemC environment (2/6)
     - Motivation of cocotb introduction
       - Encourage to do Unit Test in smaller unit, and improve efficiency and quality of Unit Test
         - Run test concurrently with SystemC coding
       - Conventional checklist-based Unit Test and TOP verification remain even if cocotb is introduced

CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**
   - Build cocotb-SystemC environment (3/6)
     - cocotb-SystemC environment



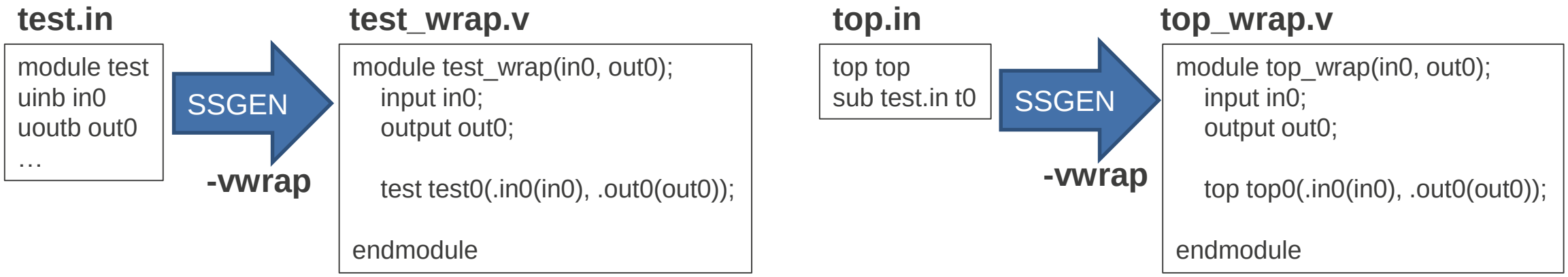
- Prepare Verilog wrapper module for top module of SystemC DUT
- Compile Verilog wrapper file and SystemC files by VCS/IES Co-simulation feature

CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**
   - Build cocotb-SystemC environment (4/6)
     - Add new command line option **-vwrap**
       - When **-vwrap** is specified, SSGEN outputs Verilog wrapper module file.
         - File name: *xxx*_wrap.v (“module test”->test_wrap.v, “top top”->top_wrap.v)
         - Module name: *xxx*_wrap (“module test”->test_wrap, “top top”->top_wrap)
         - Verilog wrapper module has same input and output ports with SystemC module
         - Verilog wrapper module instantiates SystemC module

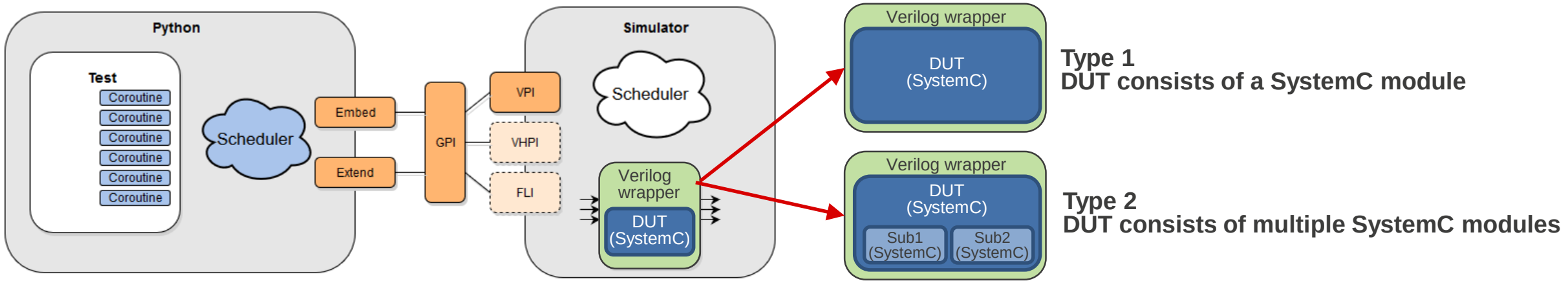
CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**
   - Build cocotb-SystemC environment (5/6)
     - Add new command line option **-vwrap** (Cont.)



CONTENTS OF THIS ACTIVITY

1. **Improve feature of verification environments generation**
   - Build cocotb-SystemC environment (6/E)
     - Modify Makefile of cocotb to compile Verilog wrapper module file and SystemC files
       - VCS: cocotb-master/makefiles/simulators/Makefile.vcs
       - IES: cocotb-master/makefiles/simulators/Makefile.ius
     - With two types of SystemC DUT, confirm compile and simulation on cocotb with your Makefile.



CONTENTS OF THIS ACTIVITY

2. Other enhancements

- Delete W003 message
  - It became unnecessary message in according with current CtoS usage.

- Prepare command-line option to not stop SSGEN due to ERROR.
  - One user wants SSGEN to report all of ERROR for .in file at a time as much as possible.
  - In default, SSGEN stops immediately due to ERROR.
  - When new command-line option is specified, SSGEN does not stop due to ERROR.

CONTENTS OF THIS ACTIVITY

3. SSGEN implement

- Implement SSGEN in according with Step1.

CONTENTS OF THIS ACTIVITY

4. SSGEN test

- Make checklist about verification environments generation improvements
- Make test data
- Import test data that you created in 2018Q1 to regression environment
- Test SSGEN

CONTENTS OF THIS ACTIVITY

5. **Update SSGEN user guide**
   - Update “ssgen_user_manual_En.doc”
     - Add enhancement information of this term.

DELIVERABLE

- Specification document, SSGEN script, SSGEN checklist and SSGEN user guide
  - **specification document**: specification for enhancements (Step1)
  - **SSGEN**: Perl script (Step3)
  - **SSGEN checklist**: checklist for enhancements (Step4)
  - **SSGEN user guide**: SSGEN user guide (Step5)

www.renesas.com