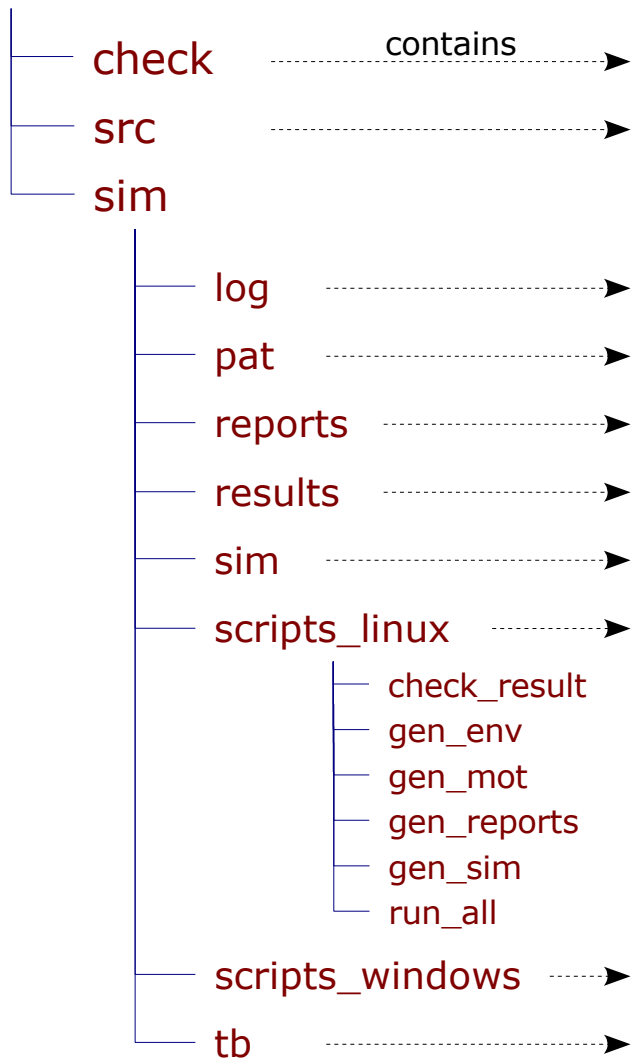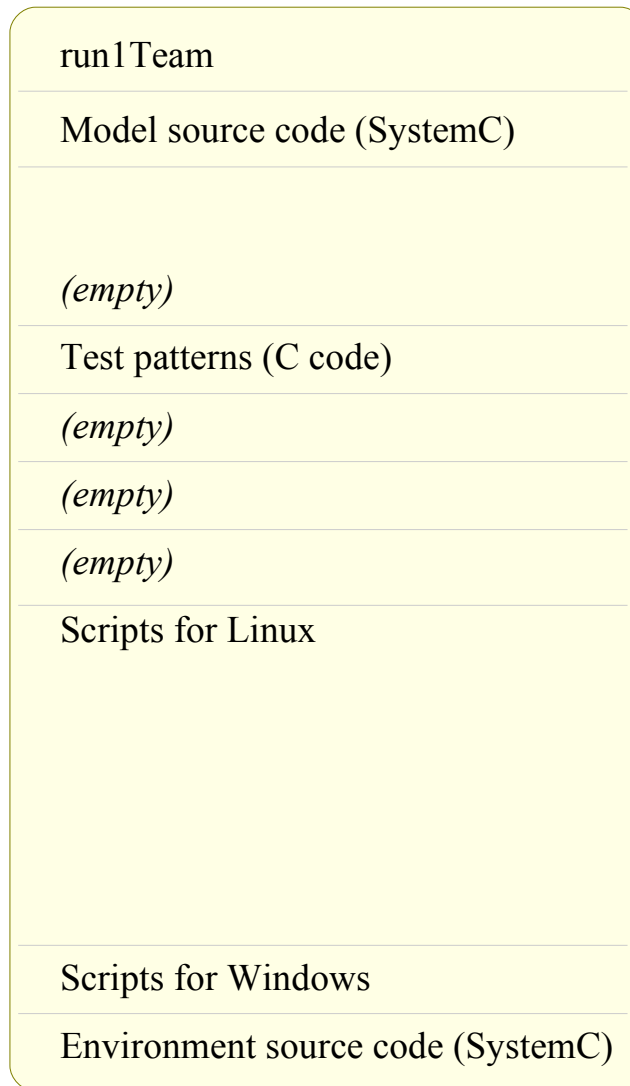# SLD environment

RVC/SLD

Outline

- Data structure
- Environment compilation
  - Model source code
- Verification with Test patterns
- Verification on Windows
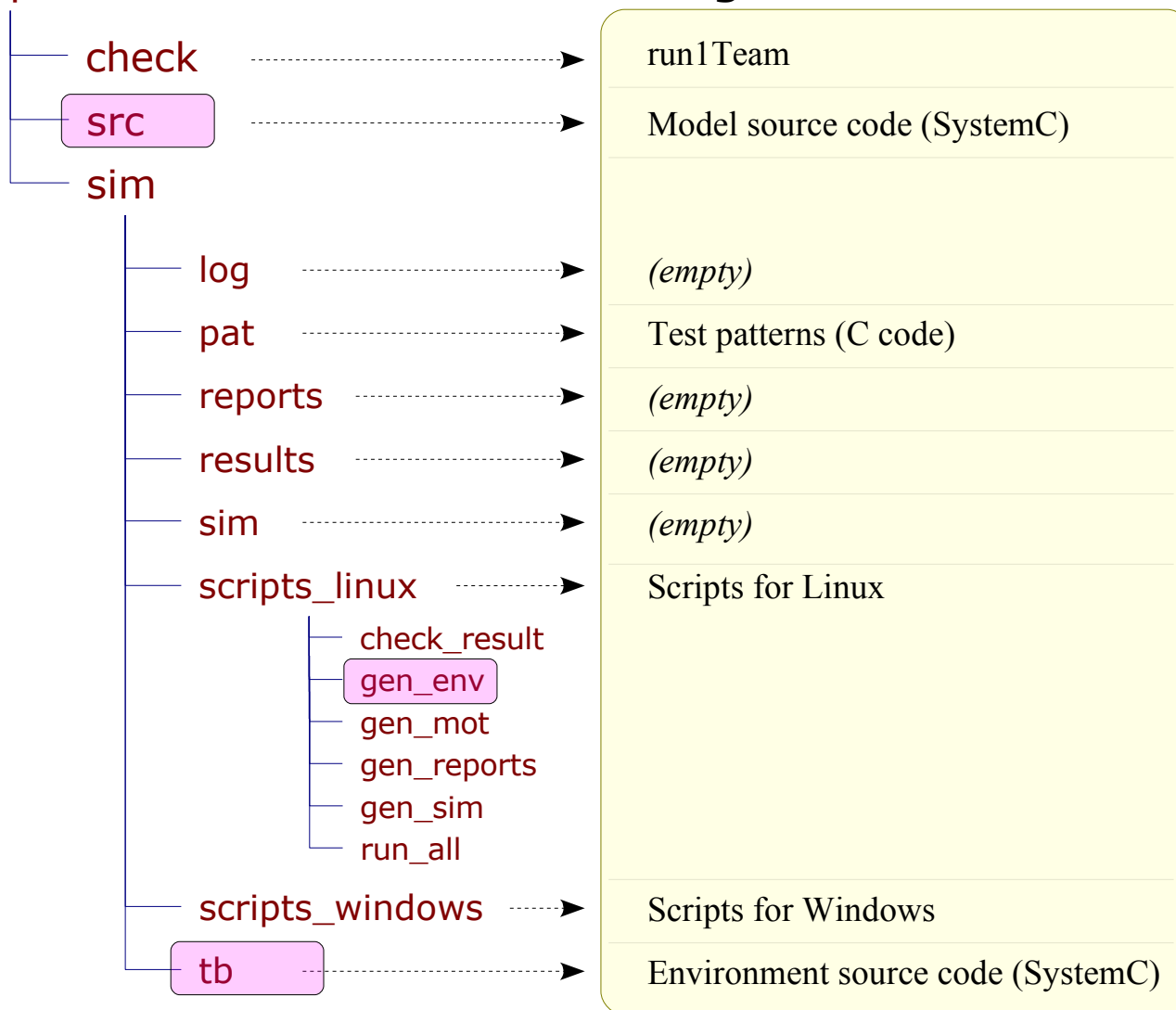
Data structure

Output

- check ········ contains ·······► run1Team
- src ·································► Model source code (SystemC)
- sim
  - log ·································► *(empty)*
  - pat ·································► Test patterns (C code)
  - reports ·······················► *(empty)*
  - results ·························► *(empty)*
  - sim ·································► *(empty)*
  - scripts_linux ············► Scripts for Linux
    - check_result
    - gen_env
    - gen_mot
    - gen_reports
    - gen_sim
    - run_all
  - scripts_windows ·······► Scripts for Windows
  - tb ·································► Environment source code (SystemC)

Original

RENESAS

Environment compilation

Output

- check ----------→ run1Team
- **src** ----------→ Model source code (SystemC)
- sim
  - log ----------→ *(empty)*
  - pat ----------→ Test patterns (C code)
  - reports ----------→ *(empty)*
  - results ----------→ *(empty)*
  - sim ----------→ *(empty)*
  - scripts_linux ----------→ Scripts for Linux
    - check_result
    - **gen_env**
    - gen_mot
    - gen_reports
    - gen_sim
    - run_all
  - scripts_windows ----→ Scripts for Windows
  - **tb** ----------→ Environment source code (SystemC)

Original

Steps

Copy model source code to environment (src → tb)

↓

gen_env: generate environment (→ tb)

↓

Compile environment (g++, Makefile)

RENESAS

Environment compilation

top.h

cpu.h

sbsc.h

Shwy.h

....

sci3.h

control_port.h

top_main.cpp

sc_main()

sci3.cpp

control_port.cpp

SystemC & TLM

compile

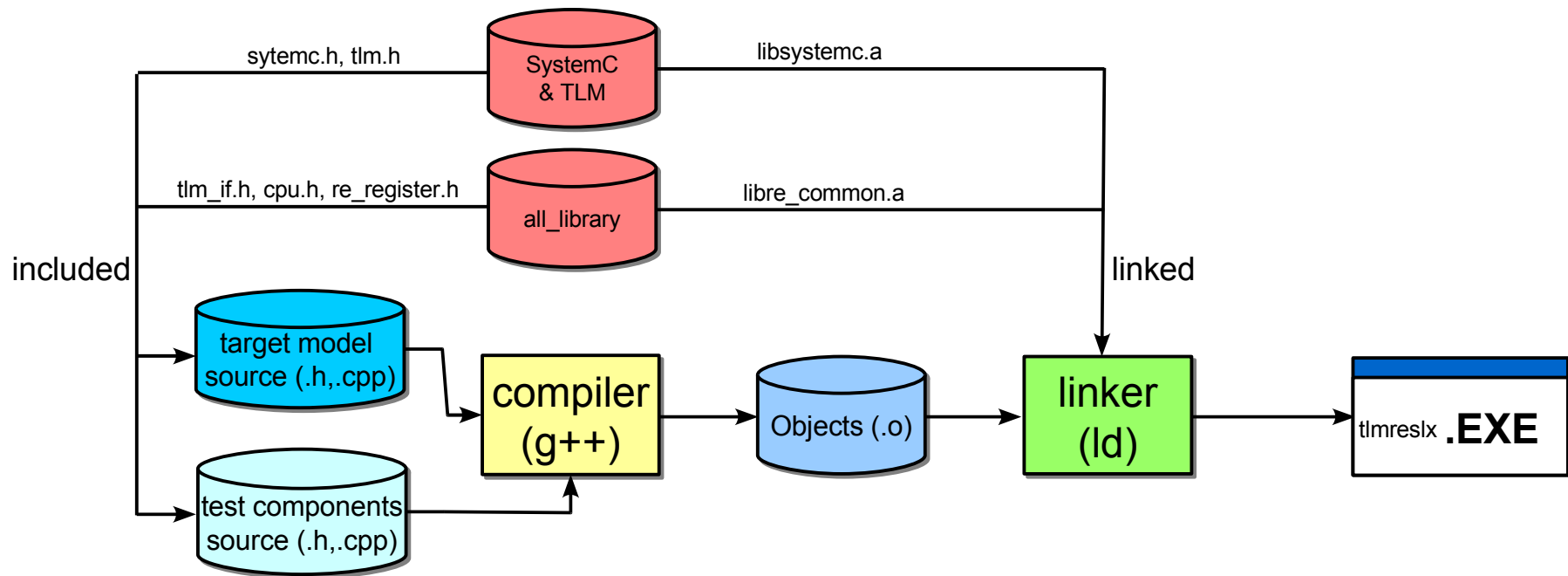tlmreslx.exe
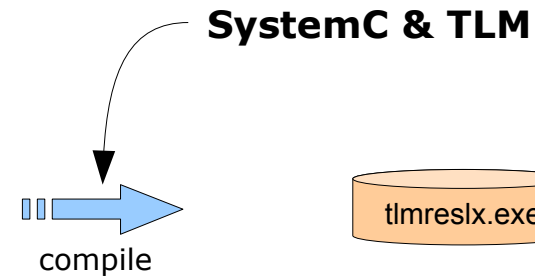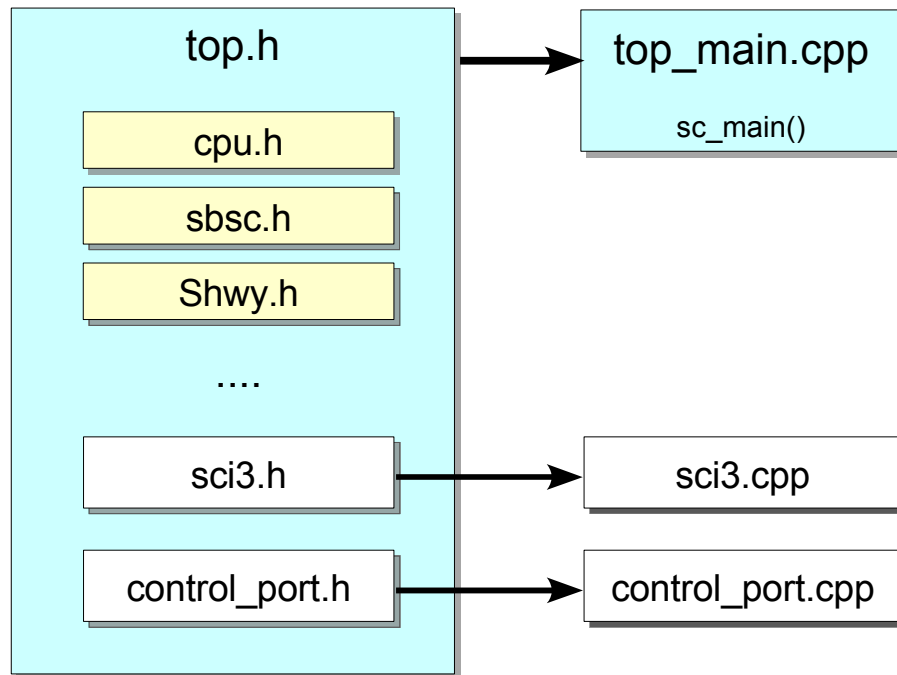
M40PF/SCI3 model is used as example
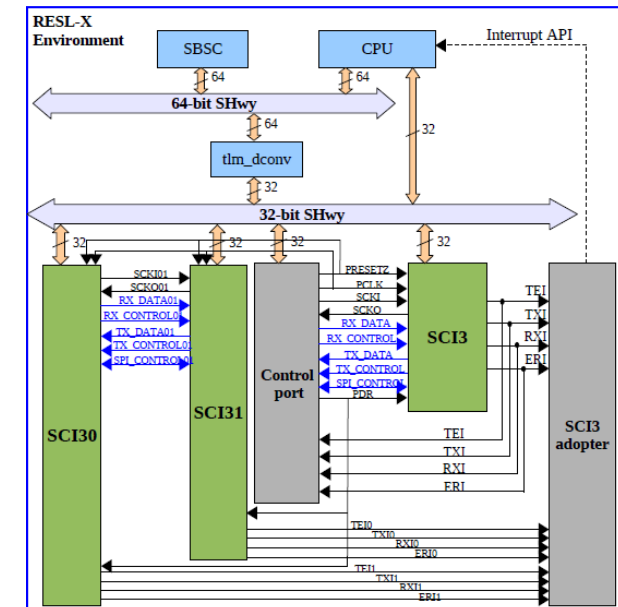
Common models → **all_library**

Generated

| A | B | File A includes file B |

| A → B | The prototype were declared in the file A will be defined in the file B |

RENESAS

Compilation process

Model source code

top.h
- cpu.h
- sbsc.h
- Shwy.h
- ....
- sci3.h
- control_port.h

top_main.cpp

sci3.cpp

control_port.cpp

sci3.h
- tlm_tgt_if.h
  - tlm_if.h
  - tlm_tgt_if_unit.h
    - tlm_if_command.h
      - re_security.h
- general_sci.h
- sci3_cmdif.h
- sci3_regif.h
  - re_register.h
    - reg_super.h
- sci3_data.h

sci3.cpp

re_security.cpp

general_sci.cpp

sci3_regif.cpp

re_register.cpp

sci3_data.cpp

Common models ———▶ **all_library**

Generated

| A | B | File A includes file B

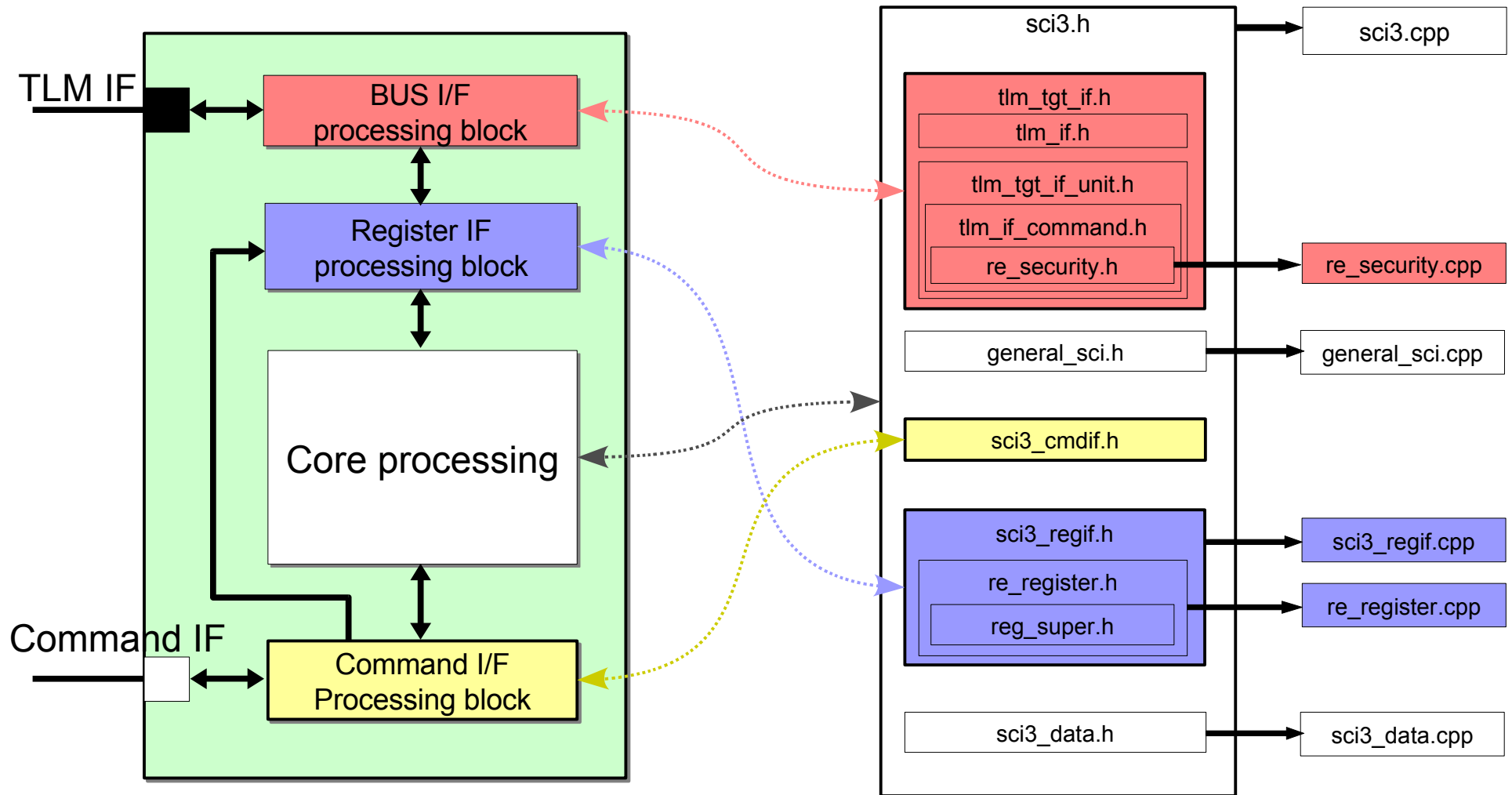A —▶ B The prototype were declared in the file A will be defined in the file B

RENESAS

Model structure

Verification with Test patterns (Linux)

Output

Original

- check ·····························▶ run1Team
- src ·····························▶ Model source code (SystemC)
- sim
  - log ·····························▶ (empty)
  - pat ·····························▶ Test patterns (C code)
  - reports ·····························▶ (empty)
  - results ·····························▶ (empty)
  - sim ·····························▶ (empty)
  - scripts_linux ·····························▶ Scripts for Linux
    - check_result
    - gen_env
    - gen_mot
    - gen_reports
    - gen_sim
    - run_all
  - scripts_windows ·····························▶ Scripts for Windows
  - tb ·····························▶ Environment source code (SystemC)

Steps

gen_mot: compile test patterns (SHCompiler) (→ pat)

↓

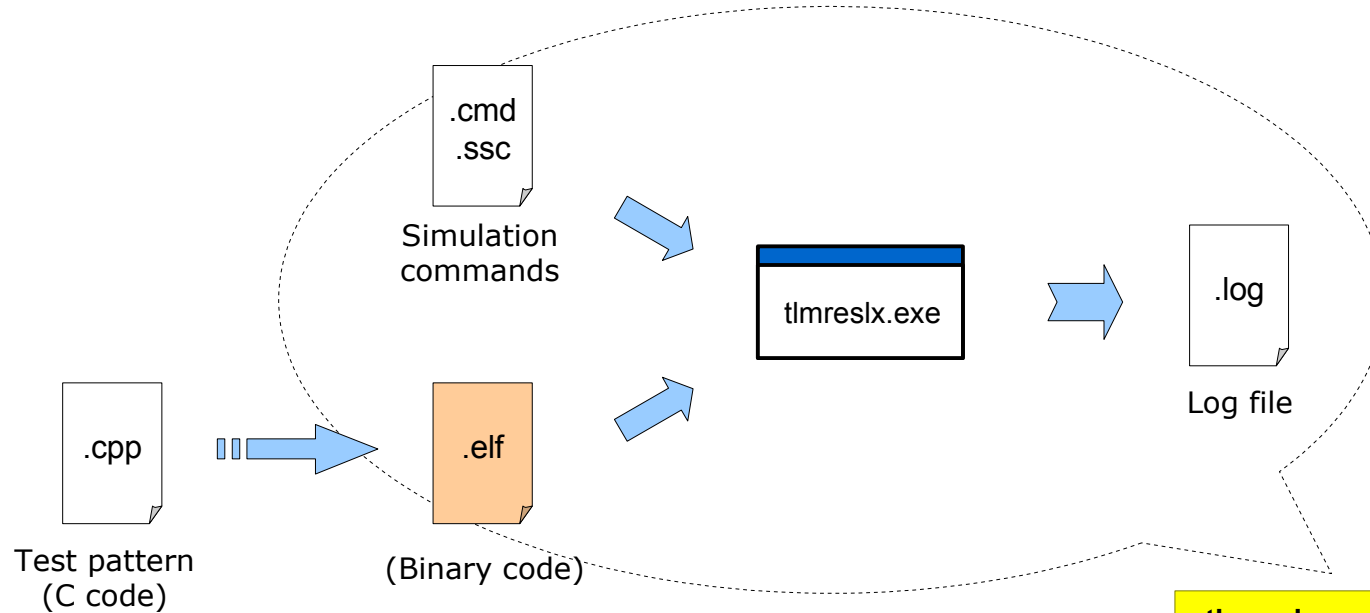gen_sim: generate scripts to run test pattern (→ sim)

↓

sim: execute test patterns, dump log files (→ log), check Pass/Fail (→ results)

↓

gen_reports: create reports (→ reports)

RENESAS

Details of running a test pattern

.cmd
.ssc

Simulation commands

tlmreslx.exe

.log

Log file

.cpp

Test pattern (C code)

.elf

(Binary code)

```c
int main()
{
    cpu_setup();
    delay(5);
    CONTROL_PORT.PCLK0.LONG = 0x1DCD6400; // 500MHz
    CONTROL_PORT.PCLK1.LONG = 0x0;
    CONTROL_PORT.HDLECOMM.BIT.FREQ_PCLK_SET  = 0x1;

    // Checking register ECMMESET
    // Checking initial value
    if ((ECM_M.ECMmESET.LONG & 0xFF) != 0x00
       || ECM_M.ECMmESSTR2.BIT.ECMmSSE230 != 0) {
         fail_bp();
    }
....
    pass_bp();
}
```

```
MOV XXXXX XXX
ADD XXXXX
EQU XXXX XXXX
JMP 0C700100
EQU XXXXX XX
MOV XXXXXX X
MOV XXXX XXX
EQU XXX XXXX
....
....
....
JMP 0C700000
```

(Assembly code is for illustration only)

```
[610000 ps] REG [ECMmESET] R Size= 1
Addr= 0xFF6B1000 Data= 0x0

[610000 ps] REG [ECMmESET:ECMmEST] R
Size= 1 Addr= 0xFF6B1000 Data= 0x0

[618004 ps] REG [ECMmESSTR2] R Size= 4
Addr= 0xFF6B0010 Data= 0x40000000
....
....
....
C0: <PC Break at H'0C700000 on CPU#0>
```
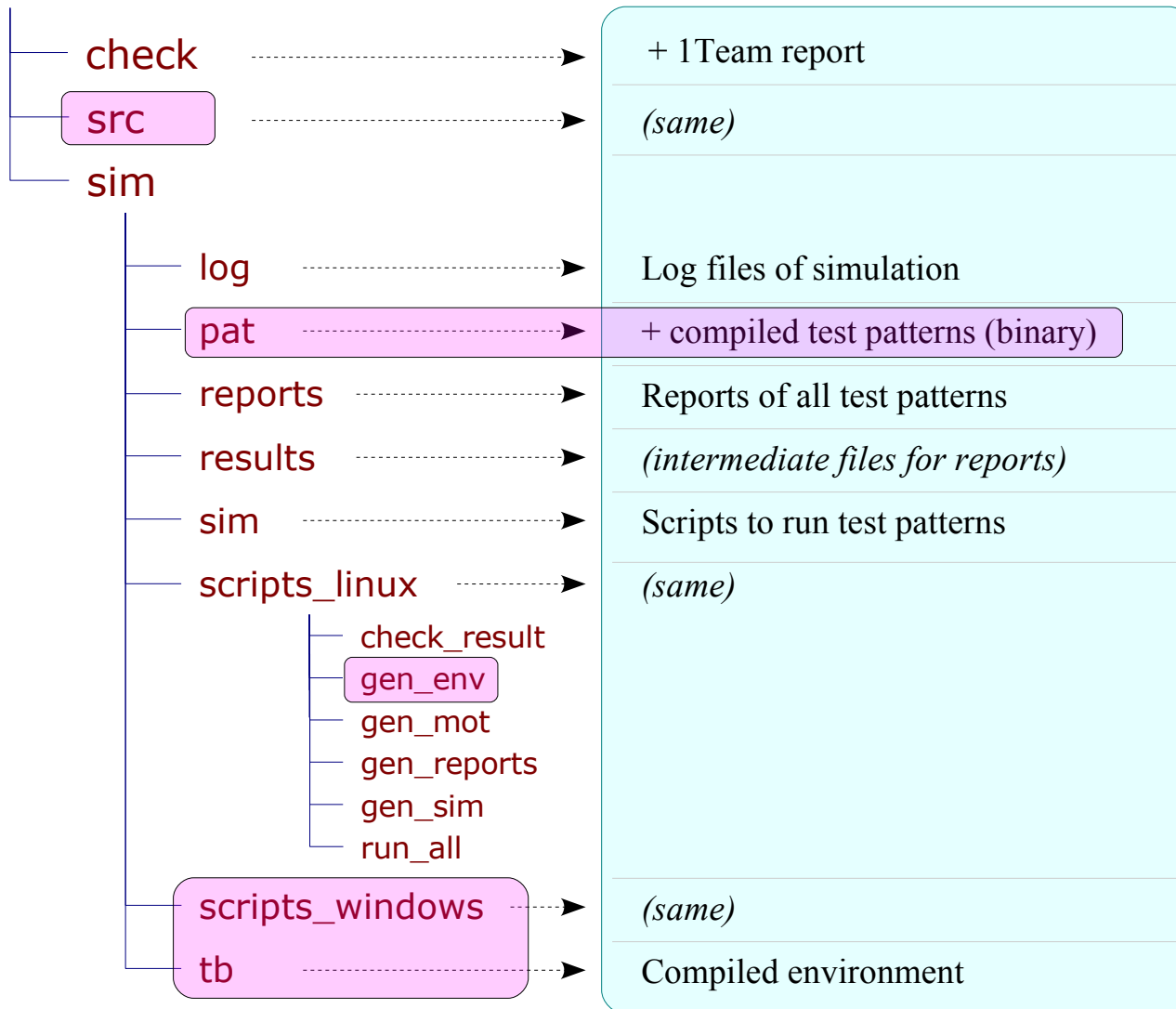
(log file)

RENESAS

Summary (Linux)

Output	Original	Executed (Linux)
check	run1Team	+ 1Team report
src	Model source code (SystemC)	*(same)*
sim		
— **log**	*(empty)*	Log files of simulation
— **pat**	Test patterns (C code)	+ compiled test patterns (binary) `gen_mot`
— **reports**	*(empty)*	Reports of all test patterns `gen_reports`
— **results**	*(empty)*	*(Results files)* `check_result`
— **sim**	*(empty)*	Scripts to run test patterns `gen_sim`
— **scripts_linux**	Scripts for Linux	*(same)*
— check_result		
— gen_env		
— gen_mot		
— gen_reports		
— gen_sim		
— run_all		
— **scripts_windows**	Scripts for Windows	*(same)*
— **tb**	Environment source code (SystemC)	Compiled environment `gen_env`

RENESAS

Verification on Windows

Output

Executed (Linux)

- check ┈┈┈┈┈┈┈┈► + 1Team report
- src ┈┈┈┈┈┈┈┈► *(same)*
- sim
  - log ┈┈┈┈┈┈┈┈► Log files of simulation
  - pat ┈┈┈┈┈┈┈┈► + compiled test patterns (binary)
  - reports ┈┈┈┈┈┈┈┈► Reports of all test patterns
  - results ┈┈┈┈┈┈┈┈► *(intermediate files for reports)*
  - sim ┈┈┈┈┈┈┈┈► Scripts to run test patterns
  - scripts_linux ┈┈┈┈┈┈┈┈► *(same)*
    - check_result
    - gen_env
    - gen_mot
    - gen_reports
    - gen_sim
    - run_all
  - scripts_windows ┈┈┈┈┈┈► *(same)*
  - tb ┈┈┈┈┈┈┈┈► Compiled environment

Steps

Copy data to Windows machine
(src, tb, pat, scripts_windows,
scripts_linux/gen_env)

↓

compile.bat: compile environment
(VC++) (→ tb)

↓

run.bat: run all test patterns,
dump log files (→ log)

↓

check_results.bat: check Pass/Fail
(→ results)

RENESAS

RENESAS

Renesas Design Vietnam Co., Ltd.