

# Lab 1 Report

Hao Xu

---

## 1. Spike model and co-sim

### 1.1 Spike model (with Risc-V pk) execution correctness

The program for test is a simple C program that calculates the sum from 1 to 9.

```
1 //
2 //  main.c
3 //  test
4 //
5 //  Created by 许昊 on 2022/10/14.
6 //
7
8 #include <stdio.h>
9
10 int main(int argc, const char * argv[]) {
11     int a = 0;
12     for ( int i = 0; i < 10; ++i){
13         a += i;
14     }
15     printf("%d\n",a);
16     return 0;
17 }
```

The result using Spike with PK is as follows:

```
[xuhao@server1 workspace]$ spike $PK test
bbl loader
45
[xuhao@server1 workspace]$
```

## 1.2 GreenRio core RTL execution correctness

Load the hex elf file in the memory:

```
dpram64 #(
    .SIZE(1<<16), // byte
    .mem_clear(1'b1),
    .memfile("isatest/rv64ui-p-andi.hex")
) magicmemory(
    .clk(clk),
    .we(s1_we),
    .din(s1_wdat),
    .waddr(s1_adr),
    .raddr(s1_adr),
    .dout(s1_rdat)
);
```

Simulate:

```
1  RTLPATH = ../rtl/lab_1/src \
2      ../rtl/lab_1/src/cache \
3      ../rtl/lab_1/src/cache/cacheblock \
4      ../rtl/lab_1/src/core_empty \
5      ../rtl/lab_1/src/core_empty/lsu \
6      ../rtl/lab_1/src/core_empty/pipeline \
7      ../rtl/lab_1/src/core_empty/units \
8      .
9
10 RTLFILE += $(foreach p,$(RTLPATH),$(wildcard $(p)/*.v $(p)/*.sv))
11 CFILE = plug.cpp
12
13 VCSFLAG = -full64 +v2k -kdb -debug_access+all +define=DUMPON+CPLUG \
14     -sverilog +vcs+lic+wait +vc+list +vpi \
15     -timescale=1ns/1ps -v2k_generate +incdir+../rtl/lab_1/src
16
17
18 compile:
19     vcs $(VCSFLAG) $(RTLFILE) $(CFILE) -top tb_hehe
```

```
Parsing design file './tb_hehe.v'
Parsing design file './dpram64.sv'
Parsing design file './wb_arb.sv'
Parsing design file './wb_interconnect.sv'
Top Level Modules:
    tb_hehe
TimeScale is 1 ns / 1 ps
Starting vcs inline pass...
```

```
8 modules and 0 UDP read.
recompiling module l1dcache
recompiling module l1licache_32
recompiling module backend
recompiling module counter_rob
recompiling module csr
recompiling module tb_hehe
recompiling module dpram64
recompiling module wb_interconnect
All of 8 modules done
```

```
make[1]: Entering directory '/work/stu/xuhao/workspace/CPULAB/riosclass_template/verilog/NewTop/csrc'
make[1]: Leaving directory '/work/stu/xuhao/workspace/CPULAB/riosclass_template/verilog/NewTop/csrc'
make[1]: Entering directory '/work/stu/xuhao/workspace/CPULAB/riosclass_template/verilog/NewTop/csrc'
```

ii. You need to have GreenRio run the risc-v elf, the corresponding results: register states, val

iii. Spike + RTL co-sim: transaction-level co-simu

i. Front-end co-sim: use the same ISA/prog decode phase (spike and RTL) - 10%

ii. Back-end co-sim: compare commit order values, pc values, etc. (spike and RTL) - 10%

## 2. Open EDA flow - 40%

a. GreenRio core logic synthesis by yosys (use 2 libs

Run the simulation result:

```
[xuhao@server1 NewTop]$ ./simv
```

```
Chronologic VCS simulator copyright 1991-2020
Contains Synopsys proprietary information.
```

```
Compiler version Q-2020.03-SP2_Full64; Runtime version Q-2020.03-SP2_Full64; Oct 25 21:57 2022
```

```
Preloading tb_hehe.magicmemory from isatest/rv64ui-p-andi.hex
VCD+ Writer Q-2020.03-SP2_Full64 Copyright (c) 1991-2020 by Synopsys Inc.
```

```
*Verdi* Loading libsscore_vcs202003.so
```

```
FSDB Dumper for VCS, Release Verdi_Q-2020.03-SP2, Linux x86_64/64bit, 08/31/2020
```

```
(C) 1996 - 2020 by Synopsys, Inc.
```

```
*Verdi* : Create FSDB file 'novas.fsdb'
```

```
*Verdi* : Begin traversing the scopes, layer (0).
```

```
*Verdi* : Enable +all dumping.
```

```
*Verdi* : End of traversing.
```

```
*Verdi* : fsdbDumpon - All FSDB files at 0 ps.
```

```
0x00000000
```

```
0x80000000
```

```
0x80000048
```

```
x1 <- 0
```

```
0x8000004C
```

```
x2 <- 0
```

i. Front-end co-sim: use the same ISA/program decode phase (spike and RTL) - 10%

ii. Back-end co-sim: compare commit order, reg values, pc values, etc. (spike and RTL) - 10%

## 2. Open EDA flow - 40%

a. GreenRio core logic synthesis by yosys (use 2 libs as d

b. GreenRio core gate level netlist + Spike co-sim execu

## 3. Presentation and Q&A - 10%

a. Pre and Q&A - 10%

```
x31 <- 11
```

```
0x8000001C
```

```
0x8000003C
```

```
x30 <- -2147479492
```

```
pass test
```

```
$finish called from file './tb_hehe.v', line 119.
```

```
$finish at simulation time 18055000
```

```
VCS Simulation Report
```

```
Time: 18055000 ps
```

```
CPU Time: 0.920 seconds;
```

```
Data structure size: 0.1Mb
```

```
Tue Oct 25 21:57:27 2022
```

```
[xuhao@server1 NewTop]$
```

## Spike and Co-sim

In lab1, you need install spike and make it work correct.

you should clone lab repo from [here](#), and configure it codes can run basic tests: ISA test and RISC-V Torture in VCS s

Finally, you need to use spike to build a co-sim framework for R

## 1.3 Spike model + GreenRio RTL co-simulation system

### 1.3.1 Run the elf file on the spike emulator and print the log

The result with log is as follows:

```
[xuhao@server1 elf]$ spike -l rv64ui-p-andi > 1.log
core 0: 0x00000000000001000 (0x00000297) auipc t0, 0x0
core 0: 0x00000000000001004 (0x02028593) addi a1, t0, 32
core 0: 0x00000000000001008 (0xf1402573) csrr a0, mhartid
core 0: 0x0000000000000100c (0x0182b283) ld t0, 24(t0)
core 0: 0x00000000000001010 (0x00028067) jr t0
core 0: 0x00000000000000000 (0x0480006f) j pc + 0x48
core 0: 0x00000000000000048 (0x00000093) li ra, 0
core 0: 0x0000000000000004c (0x00000113) li sp, 0
core 0: 0x00000000000000050 (0x00000193) li gp, 0
core 0: 0x00000000000000054 (0x00000213) li tp, 0
core 0: 0x00000000000000058 (0x00000293) li t0, 0
core 0: 0x0000000000000005c (0x00000313) li t1, 0
core 0: 0x00000000000000060 (0x00000393) li t2, 0
core 0: 0x00000000000000064 (0x00000413) li s0, 0
core 0: 0x00000000000000068 (0x00000493) li s1, 0
core 0: 0x0000000000000006c (0x00000513) li a0, 0
```

### 1.3.2 Have GreenRio run the risc-v elf in an RTL simulation environment and get the corresponding results

## 2. Open EDA flow

### 2.1 GreenRio core logic synthesis by yosys

Configure the OpenLane environment and test:

```
[xuhao@server1 OpenLane]$ make test
cd /work/stu/xuhao/tools/OpenLane && \
    docker run --rm -v /work/stu/xuhao/tools/OpenLane:/openlane -v /work/stu/xuhao/tools/OpenLane/designs:/openlane/install -v /work/stu/xuhao/tools/OpenLane/pdks -e PDK=sky130A --user 10215:1001 -e DISPLAY= -v /tmp/.X11-unix:/tmp/.X11-unix -v /work/stu/xuhao/.Xauthority:/work/stu/xuhao/.Xauthority:ro openlane_test -overwrite"
OpenLane e3a5189a1b0fc4290686fcf2ae46cd6d7947cf9f
All rights reserved. (c) 2020-2022 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[INFO]: Using configuration in 'designs/spm/config.json'...
[INFO]: PDK Root: /work/stu/xuhao/tools/OpenLane/pdks
[INFO]: Process Design Kit: sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO]: Run Directory: /openlane/designs/spm/runs/openlane_test
[INFO]: Removing existing /openlane/designs/spm/runs/openlane_test...
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[STEP 1]
[INFO]: Running Synthesis (log: designs/spm/runs/openlane_test/logs/synthesis/1-synthesis.log)...

[INFO]: There are no setup violations in the design at the typical corner.
[SUCCESS]: Flow complete.
[INFO]: Note that the following warnings have been generated:
[WARNING]: Current core area is too small for a power grid. The power grid will be minimized.

Basic test passed
```

GreenRio core logic synthesis:

```
@OpenLane Container (e3a5189):/openlane$ ./flow.tcl -design hehe
OpenLane e3a5189a1b0fc4290686fcf2ae46cd6d7947cf9f
All rights reserved. (c) 2020-2022 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[INFO]: Using configuration in 'designs/hehe/config.tcl'...
[INFO]: PDK Root: /work/stu/xuhao/tools/OpenLane/pdks
[INFO]: Process Design Kit: sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO]: Run Directory: /openlane/designs/hehe/runs/RUN_2022.10.26.14.33.51
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[STEP 1]
[INFO]: Running Synthesis (log: designs/hehe/runs/RUN_2022.10.26.14.33.51/logs/synthesis/1-synthesis.log)...
```

Use a different lib:

```
tools > OpenLane > pdks > sky130A > libs.tech > openlane > sky130_fd_sc_hd > config.tcl
1 set current_folder [file dirname [file normalize [info script]]]
2 # Technology lib
3
4 set ::env(LIB_SYNTH) "${::env(PDK_ROOT)}/${::env(PDK)}/libs.ref/${::env(STD_CELL_LIBRARY)}/lib/sky130_fd_sc_hd_tt_100C_1v80.lib"
5 set ::env(LIB_FASTEST) "${::env(PDK_ROOT)}/${::env(PDK)}/libs.ref/${::env(STD_CELL_LIBRARY)}/lib/sky130_fd_sc_hd_ff_100C_1v95.lib"
6 set ::env(LIB_SLOWEST) "${::env(PDK_ROOT)}/${::env(PDK)}/libs.ref/${::env(STD_CELL_LIBRARY)}/lib/sky130_fd_sc_hd_ss_100C_1v40.lib"
```

## 2.2 GreenRio core gate level netlist + Spike co-sim execution correctness

Get the gate level netlist file

(the file path is as follows: RUN\_2022.10.26\_15.05.51/results/synthesis/hehe.v)

```
tools > OpenLane > designs > hehe > runs > RUN_2022.10.26_15.05.51 > results > synthesis > hehe.v
1  /* Generated by Yosys 0.20+70 (git sha1 6e907acf86d, gcc 8.3.1 -fPIC -Os) */
2
3  module hehe(clk, reset, meip, m2_wbd_dat_o, m2_wbd_adr_o, m2_wbd_sel_o, m2_wbd_bl_o, m2_wbd_bry_o, m2_wbd_we_o, m2_wbd_cyc
4  , m3_wbd_stb_o, m3_wbd_dat_i, m3_wbd_ack_i, m3_wbd_lack_i, m3_wbd_err_i);
5      wire _00000;
6      wire _00001;
7      wire _00002;
8      wire _00003;
9      wire _00004;
```