

- The screenshot shows the Nova's software interface. At the top, there are tabs for 'Activities' and 'Novas'. The main menu bar includes File, View, OneTrace, Simulation, Tool, Window, and Help. Below the menu is a toolbar with icons for opening files, saving, zooming, and other functions. A status bar at the bottom displays the file path: '1\novave 2> /home/shikha/ISOC-Assignment/project/vrise-v20241209-V1.0_group5/pre_sim_single203.fsb' and various simulation parameters like '0 0 0 1ps 0 17,560,000 17,560,000 17,560,000'. The central workspace shows a waveform diagram with multiple channels and a cursor highlighting a specific point.

观察程序计数 PC 指针

参考资料

[SYNOPSYS VCS常用命令使用详解 - wenjian07的日志 - EETOP 创芯网论坛](#)

记一次VCS报错： /usr/bin/ld: undefined reference to pthread_yield

Lab3: pre_sim 例化存储器 IP

文件准备

```
● shikai@shikai-ubuntu2204:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/pre_sim$ tree -L 2
./
├── e203_filelist.f
└── riscv-tools
    ├── build.common
    ├── build-e203-rvtests.sh
    ├── fpga_test4sim
    └── README.md
        └── riscv-tests
└── rtl
    └── e203
└── run_vcs.csh
└── run_verdi.csh
└── tb
    └── tb_top.v

6 directories, 7 files
```

Lab3 工程目录结构

- ```
The final x3 Reg value: 1
TEST_PASS
#####
####
#####
#####
##
#
#####
$finish called from file "/tb/tb_top.v", line 222
$finish at simulation time 105722000
Simulation complete, time is 105722000 ps.
```

## LAB4: pre\_sim 增加 full\_chip.v

```
module ralz_chip(
 // 晶体管高速时钟 (16MHz)
 input hfxextclk,
 // 晶体管高速时钟使能信号
 output hfxoscen,
 // 晶体管低速时钟 (32.768KHz)
 input lfxextclk,
 // 晶体管低速时钟使能信号
 output lfxoscen,
 // JTAG TCK 输入, 需上拉
 input io_pads_jtag_TCK_i_ival,
```

```

// JTAG TMS 输入, 需上拉
input io_pads_jtag_TMS_i_ival,
 // JTAG TDI 输入, 需上拉
input io_pads_jtag_TDI_i_ival,
 // JTAG TDO 输出, 有使能
output io_pads_jtag_TDO_o,
 // GPIO 双向端口, 有使能
input [31:0] io_pads_gpioA,
input [31:0] io_pads_gpioB,
 // QSPI0 SCK 和 CS 输出, 无使能
output io_pads_qspi0_sck_o_oval,
output io_pads_qspi0_cs_0_o_oval,
 // QSPI0 DQ 双向 I/O, 有使能, 需上拉
input io_pads_qspi0_dq_0,
input io_pads_qspi0_dq_1,
input io_pads_qspi0_dq_2,
input io_pads_qspi0_dq_3,
 // Erst 输入, 需上拉
input io_pads_aon_ernst_n_i_ival,
 // 调试模式输入, 需上拉
input io_pads_dbgmode0_n_i_ival,
input io_pads_dbgmode1_n_i_ival,
input io_pads_dbgmode2_n_i_ival,
 // BootRom 输入, 需上拉
input io_pads_bootrom_n_i_ival,
 // 唤醒输入, 需上拉
input io_pads_aon_pmu_d wakeup_n_i_ival,
 // PMU 输出, 无使能
output io_pads_aon_pmu_padrst_o_oval,
output io_pads_aon_pmu_vddpaden_o_oval
);
 wire hfxextclk_pin;
 wire hfxoscen_pin;
 wire lfxextclk_pin;
 wire lfxoscen_pin;
 wire io_pads_jtag_TCK_i_ival_pin;
 wire io_pads_jtag_TMS_i_ival_pin;
 wire io_pads_jtag_TDI_i_ival_pin;
 wire io_pads_jtag_TDO_o_oval_pin;
 wire io_pads_jtag_TDO_o_oe_pin;
 wire [31:0] io_pads_gpioA_i_ival_pin;
 wire [31:0] io_pads_gpioA_o_oval_pin;
 wire [31:0] io_pads_gpioA_o_oe_pin;
 wire [31:0] io_pads_gpioB_i_ival_pin;
 wire [31:0] io_pads_gpioB_o_oval_pin;
 wire [31:0] io_pads_gpioB_o_oe_pin;
 wire io_pads_qspi0_sck_o_oval_pin;
 wire io_pads_qspi0_cs_0_o_oval_pin;
 wire io_pads_qspi0_dq_0_i_ival_pin;
 wire io_pads_qspi0_dq_0_o_oval_pin;
 wire io_pads_qspi0_dq_0_o_oe_pin;
 wire io_pads_qspi0_dq_1_i_ival_pin;
 wire io_pads_qspi0_dq_1_o_oval_pin;
 wire io_pads_qspi0_dq_1_o_oe_pin;
 wire io_pads_qspi0_dq_2_i_ival_pin;
 wire io_pads_qspi0_dq_2_o_oval_pin;
 wire io_pads_qspi0_dq_2_o_oe_pin;
 wire io_pads_qspi0_dq_3_i_ival_pin;
 wire io_pads_qspi0_dq_3_o_oval_pin;
 wire io_pads_qspi0_dq_3_o_oe_pin;
 wire io_pads_aon_ernst_n_i_ival_pin;
 wire io_pads_dbgmode0_n_i_ival_pin;
 wire io_pads_dbgmode1_n_i_ival_pin;
 wire io_pads_dbgmode2_n_i_ival_pin;
 wire io_pads_bootrom_n_i_ival_pin;
 wire io_pads_aon_pmu_d wakeup_n_i_ival_pin;
 wire io_pads_aon_pmu_padrst_o_oval_pin;
 wire io_pads_aon_pmu_vddpaden_o_oval_pin;

 ///
 // 实例化 e203_soc_top 模块 //
 //
e203_soc_top u_e203_soc_top(
 .hfxextclk (hfxextclk_pin),
 .hfxoscen (hfxoscen_pin),
 .lfxextclk (lfxextclk_pin),
 .lfxoscen (lfxoscen_pin),
 .io_pads_jtag_TCK_i_ival (io_pads_jtag_TCK_i_ival_pin),
 .io_pads_jtag_TMS_i_ival (io_pads_jtag_TMS_i_ival_pin),
 .io_pads_jtag_TDI_i_ival (io_pads_jtag_TDI_i_ival_pin),
 .io_pads_jtag_TDO_o_oval (io_pads_jtag_TDO_o_oval_pin),
 .io_pads_jtag_TDO_o_oe (io_pads_jtag_TDO_o_oe_pin),
 .io_pads_gpioA_i_ival (io_pads_gpioA_i_ival_pin),
 .io_pads_gpioA_o_oval (io_pads_gpioA_o_oval_pin),
 .io_pads_gpioA_o_oe (io_pads_gpioA_o_oe_pin),
 .io_pads_gpioB_i_ival (io_pads_gpioB_i_ival_pin),
 .io_pads_gpioB_o_oval (io_pads_gpioB_o_oval_pin),
 .io_pads_gpioB_o_oe (io_pads_gpioB_o_oe_pin),
 .io_pads_qspi0_sck_o_oval (io_pads_qspi0_sck_o_oval_pin),
 .io_pads_qspi0_cs_0_o_oval (io_pads_qspi0_cs_0_o_oval_pin),
 .io_pads_qspi0_dq_0_i_ival (io_pads_qspi0_dq_0_i_ival_pin),
 .io_pads_qspi0_dq_0_o_oval (io_pads_qspi0_dq_0_o_oval_pin),
 .io_pads_qspi0_dq_0_o_oe (io_pads_qspi0_dq_0_o_oe_pin),
 .io_pads_qspi0_dq_1_i_ival (io_pads_qspi0_dq_1_i_ival_pin),
 .io_pads_qspi0_dq_1_o_oval (io_pads_qspi0_dq_1_o_oval_pin),
 .io_pads_qspi0_dq_1_o_oe (io_pads_qspi0_dq_1_o_oe_pin),
 .io_pads_qspi0_dq_2_i_ival (io_pads_qspi0_dq_2_i_ival_pin),
 .io_pads_qspi0_dq_2_o_oval (io_pads_qspi0_dq_2_o_oval_pin),
 .io_pads_qspi0_dq_2_o_oe (io_pads_qspi0_dq_2_o_oe_pin),
 .io_pads_qspi0_dq_3_i_ival (io_pads_qspi0_dq_3_i_ival_pin),
 .io_pads_qspi0_dq_3_o_oval (io_pads_qspi0_dq_3_o_oval_pin),
 .io_pads_qspi0_dq_3_o_oe (io_pads_qspi0_dq_3_o_oe_pin),
 .io_pads_aon_ernst_n_i_ival (io_pads_aon_ernst_n_i_ival_pin),
 .io_pads_dbgmode0_n_i_ival (io_pads_dbgmode0_n_i_ival_pin),
 .io_pads_dbgmode1_n_i_ival (io_pads_dbgmode1_n_i_ival_pin),
 .io_pads_dbgmode2_n_i_ival (io_pads_dbgmode2_n_i_ival_pin),
 .io_pads_bootrom_n_i_ival (io_pads_bootrom_n_i_ival_pin),
 .io_pads_aon_pmu_d wakeup_n_i_ival_pin (io_pads_aon_pmu_d wakeup_n_i_ival_pin),
 .io_pads_aon_pmu_padrst_o_oval (io_pads_aon_pmu_padrst_o_oval_pin),
 .io_pads_aon_pmu_vddpaden_o_oval (io_pads_aon_pmu_vddpaden_o_oval_pin)
);

 ///
 // 实例化 IO 单元 //
 //
// hfxextclk: 输入端口
PI u_pad_hfxextclk (
 .PAD(hfxextclk),
 .C(hfxextclk_pin)
);

 // hfxoscen: 输出端口
PO4 u_pad_hfxoscen (
 .I(hfxoscen_pin),
 .PAD(hfxoscen)
);

 // lfxextclk: 输入端口
PI u_pad_lfxextclk (
 .PAD(lfxextclk),
 .C(lfxextclk_pin)
);

 // lfxoscen: 输出端口
PO4 u_pad_lfxoscen (
 .I(lfxoscen_pin),
 .PAD(lfxoscen)
);

 // io_pads_jtag_TCK_i_ival: 输入端口, 需上拉
PIU u_pad_io_pads_jtag_TCK_i_ival (
 .PAD(io_pads_jtag_TCK_i_ival),
 .C(io_pads_jtag_TCK_i_ival_pin)
);

 // io_pads_jtag_TMS_i_ival 和 io_pads_jtag_TDI_i_ival: 输入端口, 需上拉
PIU u_pad_io_pads_jtag_TMS_i_ival (
 .PAD(io_pads_jtag_TMS_i_ival),
 .C(io_pads_jtag_TMS_i_ival_pin)
);

 // io_pads_jtag_TDI_i_ival: 输入端口, 需上拉
PIU u_pad_io_pads_jtag_TDI_i_ival (
 .PAD(io_pads_jtag_TDI_i_ival),
 .C(io_pads_jtag_TDI_i_ival_pin)
);

 // io_pads_qspi0_dq_0_o_oval 和 io_pads_qspi0_dq_1_o_oval: 三态输出
POT16 u_pad_io_pads_qspi0_dq_0_o_oval (
 .OEN(~io_pads_qspi0_dq_0_o_oe_pin),
 .I(io_pads_qspi0_dq_0_o_oval_pin),
 .PAD(io_pads_qspi0_dq_0_o_oval)
);

 // io_pads_qspi0_dq_1_o_oval 和 io_pads_qspi0_dq_2_o_oval: 三态输出
POT16 u_pad_io_pads_qspi0_dq_1_o_oval (
 .OEN(~io_pads_qspi0_dq_1_o_oe_pin),
 .I(io_pads_qspi0_dq_1_o_oval_pin),
 .PAD(io_pads_qspi0_dq_1_o_oval)
);

 // io_pads_qspi0_dq_2_o_oval 和 io_pads_qspi0_dq_3_o_oval: 三态输出
POT16 u_pad_io_pads_qspi0_dq_2_o_oval (
 .OEN(~io_pads_qspi0_dq_2_o_oe_pin),
 .I(io_pads_qspi0_dq_2_o_oval_pin),
 .PAD(io_pads_qspi0_dq_2_o_oval)
);

 // io_pads_qspi0_dq_3_o_oval 和 io_pads_qspi0_dq_0_o_oval: 三态输出
POT16 u_pad_io_pads_qspi0_dq_3_o_oval (
 .OEN(~io_pads_qspi0_dq_3_o_oe_pin),
 .I(io_pads_qspi0_dq_3_o_oval_pin),
 .PAD(io_pads_qspi0_dq_3_o_oval)
);

 // io_pads_qspi0_dq_0: 双向端口
genvar i;
generate
 for (i = 0; i < 32; i = i + 1) begin : gpioA
 PB8 u_pad_gpioAX (
 .OEN(~io_pads_gpioA_o_oe_pin[i]),
 .C(io_pads_gpioA_i_ival_pin[i]),
 .PAD(io_pads_gpioA[i]),
 .I(io_pads_gpioA_o_oval_pin[i])
);
 end
endgenerate

 // io_pads_gpioB: 双向端口
generate
 for (i = 0; i < 32; i = i + 1) begin : gpioB
 PB8 u_pad_gpioBX (
 .OEN(~io_pads_gpioB_o_oe_pin[i]),
 .C(io_pads_gpioB_i_ival_pin[i]),
 .PAD(io_pads_gpioB[i]),
 .I(io_pads_gpioB_o_oval_pin[i])
);
 end
endgenerate

 // io_pads_qspi0_sck_o_oval: 输出端口
PO16 u_pad_io_pads_qspi0_sck_o_oval (
 .I(io_pads_qspi0_sck_o_oval_pin),
 .PAD(io_pads_qspi0_sck_o_oval)
);

 // io_pads_qspi0_cs_0_o_oval: 输出端口
PO16 u_pad_io_pads_qspi0_cs_0_o_oval (
 .I(io_pads_qspi0_cs_0_o_oval_pin),
 .PAD(io_pads_qspi0_cs_0_o_oval)
);

 // io_pads_qspi0_dq_0: 双向端口
PBCU16 u_pad_io_pads_qspi0_dq_0 (
 .PAD(io_pads_qspi0_dq_0),
 .OEN(1'b0),
 .I(io_pads_qspi0_dq_0_o_oval_pin),
 .C(io_pads_qspi0_dq_0_i_ival_pin)
);

 // io_pads_qspi0_dq_1, io_pads_qspi0_dq_2, io_pads_qspi0_dq_3: 双向端口
PBCU16 u_pad_io_pads_qspi0_dq_1 (
 .PAD(io_pads_qspi0_dq_1),
 .OEN(~io_pads_qspi0_dq_1_o_oe_pin),
 .REN(1'b0),
 .I(io_pads_qspi0_dq_1_o_oval_pin),
 .C(io_pads_qspi0_dq_1_i_ival_pin)
);

 // io_pads_qspi0_dq_2 (
 .PAD(io_pads_qspi0_dq_2),
 .OEN(~io_pads_qspi0_dq_2_o_oe_pin),
 .REN(1'b0),
 .I(io_pads_qspi0_dq_2_o_oval_pin),
 .C(io_pads_qspi0_dq_2_i_ival_pin)
);

 // io_pads_qspi0_dq_3 (
 .PAD(io_pads_qspi0_dq_3),
 .OEN(~io_pads_qspi0_dq_3_o_oe_pin),
 .REN(1'b0),
 .I(io_pads_qspi0_dq_3_o_oval_pin),
 .C(io_pads_qspi0_dq_3_i_ival_pin)
);

 // io_pads_aon_ernst_n_i_ival: 输入端口, 需上拉
PIU u_pad_io_pads_aon_ernst_n_i_ival (
 .PAD(io_pads_aon_ernst_n_i_ival),
 .C(io_pads_aon_ernst_n_i_ival_pin)
);

 // io_pads_dbgmode0_n_i_ival: 输入端口, 需上拉
PIU u_pad_io_pads_dbgmode0_n_i_ival (
 .PAD(io_pads_dbgmode0_n_i_ival),
 .C(io_pads_dbgmode0_n_i_ival_pin)
);

 // io_pads_dbgmode1_n_i_ival 和 io_pads_dbgmode2_n_i_ival: 输入端口, 需上拉
PIU u_pad_io_pads_dbgmode1_n_i_ival (
 .PAD(io_pads_dbgmode1_n_i_ival),
 .C(io_pads_dbgmode1_n_i_ival_pin)
);

 // io_pads_bootrom_n_i_ival: 输入端口, 需上拉
PIU u_pad_io_pads_bootrom_n_i_ival (
 .PAD(io_pads_bootrom_n_i_ival),
 .C(io_pads_bootrom_n_i_ival_pin)
);

 // io_pads_aon_pmu_d wakeup_n_i_ival: 输入端口, 需上拉
PIU u_pad_io_pads_aon_pmu_d wakeup_n_i_ival (
 .PAD(io_pads_aon_pmu_d wakeup_n_i_ival),
 .C(io_pads_aon_pmu_d wakeup_n_i_ival_pin)
);

 // io_pads_aon_pmu_padrst_o_oval 和 io_pads_aon_pmu_vddpaden_o_oval: 输出端口
PO16 u_pad_io_pads_aon_pmu_padrst_o_oval (
 .I(io_pads_aon_pmu_padrst_o_oval),
 .PAD(io_pads_aon_pmu_vddpaden_o_oval)
);

 // io_pads_aon_pmu_vddpaden_o_oval: 输出端口
PO16 u_pad_io_pads_aon_pmu_vddpaden_o_oval (
 .I(io_pads_aon_pmu_vddpaden_o_oval),
 .PAD(io_pads_aon_pmu_vddpaden_o_oval)
);

endmodule

```

例化：

- 309  
310  
311  
312  
313  
314  
315

```
319 assign io_pads_qspi0_dq_1 = 1'b1;
320 assign io_pads_qspi0_dq_2 = 1'b1;
321 assign io_pads_qspi0_dq_3 = 1'b1;
322
323 assign gpioA = 32'b0;
324 assign gpioB = 32'b0;
325
326 full_chip u_full_chip(
327 .hfextclk (hfclk),
328 .hfxoscen (),
329 .lfextclk (lfextclk),
```

```
.io_pads_jtag_TCK_i_ival (jtag_TCK),
.io_pads_jtag_TMS_i_ival (jtag_TMS),
.io_pads_jtag_TDI_i_ival (jtag_TDI),
.io_pads_jtag_TDO_o (jtag_TDO),
```

|     |       |
|-----|-------|
| 341 | .io_p |
| 342 | .io_p |
| 343 | .io_p |
| 344 | .io_p |
| 345 | .io_p |
| 346 | .io_p |

```
// 测试模式输入，带下拉
input test_mode
```

|  |     |          |
|--|-----|----------|
|  | 98  | wire io_ |
|  | 99  | wire tes |
|  | 100 |          |

```
308 .P016_u_pad_io_pads_aon_pmu_vddpden_o_oval(
309 .I.io_pads_aon_pmu_vddpden_o_oval),
310 .O.io_pads_aon_pmu_vddpden_o_oval)
311);
312
313 // 实例化 test_mode 输入端口, 带下拉配置
314 PID_U_PAD_TEST_MODE(
315 .PAD(test_mode),
316 .C(test_mode_pin)
317);
318
319 endmodule
320
```

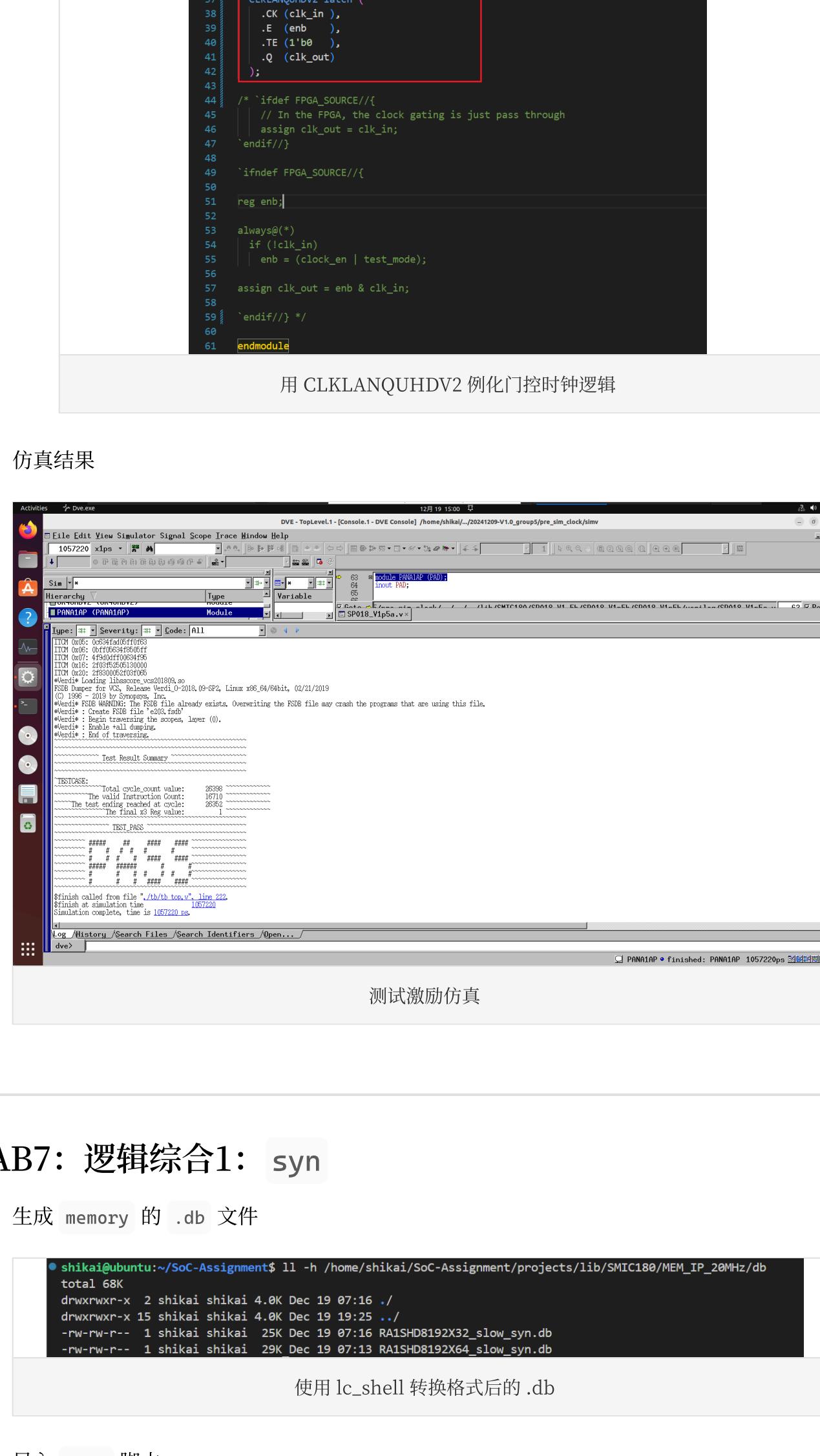
full\_chip 模块定义内, 添加 PID test\_mode 实例

- tb\_top.v 测试激励顶层添加 test\_mode 输入

```
299 //
300 //
301 wire JTAG_TDI 1'b0;
302 wire JTAG_TDO 1'b0;
303 wire JTAG_TCK 1'b0;
304 wire JTAG_TS 1'b0;
305 wire JTAG_RST 1'b0;
306 wire JTAG_RV_DIO 1'b0;
307 wire JTAG_RV_N 1'b0;
308 wire JTAG_RV_S 1'b0;
309 wire [31:0] gpios;
310 wire [31:0] gpiod;
311 wire [31:0] gpiob;
312 wire [31:0] gpioc;
313 wire [31:0] gpiod_qspi0_dq_0;
314 wire [31:0] gpiod_qspi0_dq_1;
315 wire [31:0] gpiod_qspi0_dq_2;
316 wire [31:0] gpiod_qspi0_dq_3;
317 wire test_mode = 1'b0;
318 assign io_pads_qspi0_dq_0 = 1'b1;
319 assign io_pads_qspi0_dq_1 = 1'b1;
320 assign io_pads_qspi0_dq_2 = 1'b1;
321 assign io_pads_qspi0_dq_3 = 1'b1;
322
323 assign gpiob = 32'b0;
324 assign gpiod = 32'b0;
325 assign gpioc = 32'b0;
326 assign gpiod_qspi0_dq_0 = 1'b1;
327 assign gpiod_qspi0_dq_1 = 1'b1;
328 assign gpiod_qspi0_dq_2 = 1'b1;
329 assign gpiod_qspi0_dq_3 = 1'b1;
330
331 assign gpiob_rst_n = 1'b0;
332 assign gpiod_rst_n = 1'b0;
333 assign gpioc_rst_n = 1'b0;
334 assign gpiod_qspi0_sck_o_oval = 1'b0;
335 assign gpiod_qspi0_cs_o_oval = 1'b0;
336 assign gpiod_qspi0_dq_0 = 1'b0;
337 assign gpiod_qspi0_dq_1 = 1'b0;
338 assign gpiod_qspi0_dq_2 = 1'b0;
339 assign gpiod_qspi0_dq_3 = 1'b0;
340
341 assign gpiob_mosi0 = 1'b0;
342 assign gpiob_miso0 = 1'b0;
343 assign gpiod_qspi0_sck_o_oval = 1'b0;
344 assign gpiod_qspi0_cs_o_oval = 1'b0;
345 assign gpiod_qspi0_dq_0 = 1'b0;
346 assign gpiod_qspi0_dq_1 = 1'b0;
347 assign gpiod_qspi0_dq_2 = 1'b0;
348 assign gpiod_qspi0_dq_3 = 1'b0;
349
350 assign io_pads_aon_rst_n_i_ival = 1'b0;
351 assign io_pads_dtmf_gated_n_i_ival = 1'b0;
352 assign io_pads_dtmf_gated_n_i_ival = 1'b0;
353 assign io_pads_dtmf_gated_n_i_ival = 1'b0;
354 assign io_pads_dtmf_gated_n_i_ival = 1'b0;
355
356 assign io_pads_bottom_m_i_ival = 1'b0;
357 assign io_pads_ton_pmu_dkewup_n_i_ival = 1'b0;
358 assign io_pads_aon_pmu_padrst_o_oval = 1'b0;
359 assign io_pads_aon_pmu_vddpden_o_oval = 1'b0;
360
361 test_mode = 1'b0;
362
363 y;
```

测试激励例化 full\_chip 时添加 test\_mode 端口

- 仿真通过



The screenshot shows the ModelSim simulation environment. It displays the tb\_top.v testbench code, which includes the instantiation of the full\_chip module and various signal assignments. The simulation results window shows the waveform for the test\_mode signal, which is asserted at the beginning of the simulation. The status bar indicates the simulation has finished successfully.

仿真测试 test\_mode 使能端口

## LAB6: 时钟网络分析



The screenshot shows the ModelSim simulation environment. It displays the tb\_top.v testbench code, which includes the instantiation of the full\_chip module and various signal assignments. The simulation results window shows the waveform for the test\_mode signal, which is asserted at the beginning of the simulation. The status bar indicates the simulation has finished successfully.

思考题三

- 思考题三:

上图中, 左边是RTL代码中时钟信号的相关逻辑操作, 右边是逻辑综合后对应的网表, 综合的结果符合预期吗? 为什么? 应该怎么处理?

- 答:
- 原因分析
  - 在左侧的 RTL 代码中, 时钟信号 clkout 经历了一系列逻辑操作。这些操作涉及到对时钟信号的门控, 通常是为了节省功耗或在特定条件下启用时钟。但它们可能会导致时钟的边缘失真或引入额外的延迟, 尤其是在没有使用专用的时钟单元时 (如时钟门控单元)。
  - 在右侧的网表中, 可以看到时钟信号的逻辑操作被转换为特定的逻辑门。这些门可能会影响时钟信号的完整性, 导致时钟信号的边缘变得不清晰。例如, 使用普通的逻辑门 (如与门、或门) 来处理时钟信号可能会引入额外的延迟和噪声, 导致时钟信号的质量下降。

- 解决办法

- 应该用标准单元库中的时钟专用单元 (如 clk\_gate、clk\_mux 等) 替代 RTL 中的逻辑操作。这些单元专门设计用于处理时钟信号, 能够有效降低时钟信号的延迟和噪声。在后续的布局和布线阶段, 确保进行时钟树综合, 以优化时钟信号的分发, 减少时钟偏差和抖动。

- 例化时钟专用 cell

- e203\_subsys\_gfcv.v 将时钟或操作改为用 CLKOR2UHDV4 代替

```
195 // assign clkout = clk0_gated | clk1_gated;
196 CLKOR2UHDV4 U_CLKOR2(
197 .Z (hfclk),
198 .I0 (gfcm_clk),
199 .I1 (hfextclk),
200 .S (test_mode)
201);
202
```

用 CLKOR2UHDV4 例化 clkout

- e203\_subsys\_hclkgen.v 将时钟 2选1 操作改为用 CLKMUX2UHDV4 代替

```
119 // assign hfclk = test_mode ? hfextclk : gfcv_clk;
120 CLKMUX2UHDV4 U_CLKMUX2(
121 .Z (hfclk),
122 .I0 (gfcm_clk),
123 .I1 (hfextclk),
124 .S (test_mode)
125);
126
```

用 CLKMUX2UHDV4 例化 hfclk

- e203\_clkgate.v 将门控时钟逻辑用 CLKLANQUDHV2 替代

```
26 'include "e203_defines.v"
27 module #(
28 input clk_in,
29 input test_mode,
30 input clock_en,
31 output clock_en,
32 output clk_out
33)(
34 wire enb;
35 assign enb = (clock_en | test_mode);
36 CLKLANQUDHV2 U_CLKGEN(
37 .CLK_IN(clk_in),
38 .TEST_MODE(test_mode),
39 .ENB(enb),
40 .CLK_OUT(clk_out)
41);
42
43 `ifndef FPGA_SOURCE//{
44 reg enb;
45 always@(*)
46 if (!clock_in)
47 enb = (clock_en | test_mode);
48 assign clk_out = enb & clk_in;
49 `endif///
50
51 endmodule
52
```

用 CLKLANQUDHV2 例化门控时钟逻辑

- 仿真结果



The screenshot shows the ModelSim simulation environment. It displays the tb\_top.v testbench code, which includes the instantiation of the full\_chip module and various signal assignments. The simulation results window shows the waveform for the test\_mode signal, which is asserted at the beginning of the simulation. The status bar indicates the simulation has finished successfully.

测试激励仿真

## LAB7: 逻辑综合1: syn

- 生成 memory 的 .db 文件

```
shikai@ubuntu:~/SoC-Assignment$ ll -h /home/shikai/SoC-Assignment/projects/lib/SMIC180/MEM_IP_20MHz/db
total 68K
drwxrwxr-x 2 shikai shikai 4.0K Dec 19 07:16 .
drwxrwxr-x 15 shikai shikai 4.0K Dec 19 19:25 ..
-rw-rw-r-- 1 shikai shikai 25K Dec 19 07:16 RAISHD8192X32_slow_syn.db
-rw-rw-r-- 1 shikai shikai 29K Dec 19 07:13 RAISHD8192X64_slow_syn.db
```

使用 lc\_shell 转换格式后的 .db

- 导入 .tcl 脚本

将 LAB7\_syn 下的 setup\_dc.tcl、cons.tcl 以及 dc.tcl 拷贝至 projects/risc-v/20241209-V1.0\_group5/syn/scripts。

- 通过 dc\_shell 进行逻辑综合

- 命令行执行 dc\_shell-t (忽略/bin/csh不存在的问题, 其他Linux发现版本可能内置了csh, 就不会报这个错)

- 进行 syn

- source ./scripts/dc.tcl > dc.log

报错:

Error: Invalid delay direction for port xxx

```
168 set_output_delay $OUT_DELAY_MASTER -clock hfextclk [get_ports {io_pads_gpioA io_pads_gpioB}]
169 Error: Invalid delay direction for port `io_pads_gpioA[31]` . (UID:254)
170
171 set_output_delay $OUT_DELAY_MASTER -clock hfextclk [get_ports {io_pads_qspi0_dq0_o_oval}]
172 Error: Invalid delay direction for port `io_pads_qspi0_dq0_0` . (UID:254)
173
174 set_output_delay $OUT_DELAY_MASTER -clock hfextclk [get_ports {io_pads_qspi0_dq0_1}]
175 Error: Invalid delay direction for port `io_pads_qspi0_dq0_1` . (UID:254)
176
177 set_output_delay $OUT_DELAY_MASTER -clock hfextclk [get_ports {io_pads_qspi0_dq0_2}]
178 Error: Invalid delay direction for port `io_pads_qspi0_dq0_2` . (UID:254)
179
180 set_output_delay $OUT_DELAY_MASTER -clock hfextclk [get_ports {io_pads_qspi0_dq0_3}]
181 Error: Invalid delay direction for port `io_pads_qspi0_dq0_3` . (UID:254)
```

报错: delay 方向错误

- 思考题四:

上图中, 左边是RTL代码中时钟信号的相关逻辑操作, 右边是逻辑综合后对应的网表, 综合的结果符合预期吗? 为什么? 应该怎么处理?

- 答:
- 原因分析
  - 在左侧的 RTL 代码中, 时钟信号 clkout 经历了一系列逻辑操作。这些操作涉及到对时钟信号的门控, 通常是为了节省功耗或在特定条件下启用时钟。但它们可能会导致时钟的边缘失真或引入额外的延迟, 尤其是在没有使用专用的时钟单元时 (如时钟门控单元)。
  - 在右侧的网表中, 可以看到时钟信号的逻辑操作被转换为特定的逻辑门。这些门可能会影响时钟信号的完整性, 导致时钟信号的边缘变得不清晰。例如, 使用普通的逻辑门 (如与门、或门) 来处理时钟信号可能会引入额外的延迟和噪声, 导致时钟信号的质量下降。

- 解决办法

- 应该用标准单元库中的时钟专用单元 (如 clk\_gate、clk\_mux 等) 替代 RTL 中的逻辑操作。这些单元专门设计用于处理时钟信号, 能够有效降低时钟信号的延迟和噪声。在后续的布局和布线阶段, 确保进行时钟树综合, 以优化时钟信号的分发, 减少时钟偏差和抖动。

- 例化时钟专用 cell

- e203\_subsys\_gfcv.v 将时钟或操作改为用 CLKMUX2UHDV4 代替

```
195 // assign clkout = clk0_gated | clk1_gated;
196 CLKMUX2UHDV4 U_CLKOR2(
197 .Z (hfclk),
198 .I0 (gfcm_clk),
199 .I1 (hfextclk),
200 .S (test_mode)
201);
202
```

用 CLKOR2UHDV4 例化 clkout

- e203\_subsys\_hclkgen.v 将时钟 2选1 操作改为用 CLKMUX2UHDV4 代替

```
26 'include "e203_defines.v"
27 module #(
28 input clk_in,
29 input test_mode,
30 input clock_en,
31 output clock_en,
32 output clk_out
33)(
34 wire enb;
35 assign enb = (clock_en | test_mode);
36 CLKMUX2UHDV4 U_CLKGEN(
37 .CLK_IN(clk_in),
38 .TEST_MODE(test_mode),
39 .ENB(enb),
40 .CLK_OUT(clk_out)
41);
42
43 `ifndef FPGA_SOURCE//{
44 reg enb;
45 always@(*)
46 if (!clock_in)
47 enb = (clock_en | test_mode);
48 assign clk_out = enb & clk_in;
49 `endif///
50
51 endmodule
52
```

用 CLKMUX2UHDV4 例化 hfclk

- e203\_clkgate.v 将门控时钟逻辑用 CLKLANQUDHV2 替代

```
26 'include "e203_defines.v"
27 module #(
28 input clk_in,
29 input test_mode,
30 input clock_en,
31 output clock_en,
32 output clk_out
33)(
34 wire enb;
35 assign enb = (clock_en | test_mode);
36 CLKLANQUDHV2 U_CLKGEN(
37 .CLK_IN(clk_in),
38 .TEST_MODE(test_mode),
39 .ENB(enb),
40 .CLK_OUT(clk_out)
41);
42
43 `ifndef FPGA_SOURCE//{
44 reg enb;
45 always@(*)
46 if (!clock_in)
47 enb = (clock_en | test_mode);
48 assign clk_out = enb & clk_in;
49 `endif///
50
51 endmodule
52
```

用 CLKLANQUDHV2 例化门控时钟逻辑

- 仿真结果



The screenshot shows the ModelSim simulation environment. It displays the tb\_top.v testbench code, which includes the instantiation of the full\_chip module and various signal assignments. The simulation results window shows the waveform for the test\_mode signal, which is asserted at the beginning of the simulation. The status bar indicates the simulation has finished successfully.

测试激励仿真

## LAB8: 逻辑综合2: dc.log

- 锁存器 Latch

```
projects > risc-v > 20241209-V1.0_group5 > syn > dc.log
 32 FIFO_tx_data = 0;
 33 reg [31:0] regs_n;
 34 reg [31:0] trigger_level_n;
 35 reg [31:0] trigger_level_q;
 36 reg [31:0] trigger_level_o;
 37 reg [31:0] trigger_level_d;
 38 reg [31:0] trigger_level_qd;
 39 reg [31:0] trigger_level_dq;
 40 reg [31:0] trigger_level_dq2;
 41 reg [31:0] trigger_level_dq3;
 42 reg [31:0] trigger_level_dq4;
 43 reg [31:0] trigger_level_dq5;
 44 reg [31:0] trigger_level_dq6;
 45 reg [31:0] trigger_level_dq7;
 46 reg [31:0] trigger_level_dq8;
 47 reg [31:0] trigger_level_dq9;
 48 reg [31:0] trigger_level_dq10;
 49 reg [31:0] trigger_level_dq11;
 50 reg [31:0] trigger_level_dq12;
 51 reg [31:0] trigger_level_dq13;
 52 reg [31:0] trigger_level_dq14;
 53 reg [31:0] trigger_level_dq15;
 54 reg [31:0] trigger_level_dq16;
 55 reg [31:0] trigger_level_dq17;
 56 reg [31:0] trigger_level_dq18;
 57 reg [31:0] trigger_level_dq19;
 58 reg [31:0] trigger_level_dq20;
 59 reg [31:0] trigger_level_dq21;
 60 reg [31:0] trigger_level_dq22;
 61 reg [31:0] trigger_level_dq23;
 62 reg [31:0] trigger_level_dq24;
 63 reg [31:0] trigger_level_dq25;
 64 reg [31:0] trigger_level_dq26;
 65 reg [31:0] trigger_level_dq27;
 66 reg [31:0] trigger_level_dq28;
 67 reg [31:0] trigger_level_dq29;
 68 reg [31:0] trigger_level_dq30;
 69 reg [31:0] trigger_level_dq31;
 70 reg [31:0] trigger_level_dq32;
 71 reg [31:0] trigger_level_dq33;
 72 reg [31:0] trigger_level_dq34;
 73 reg [31:0] trigger_level_dq35;
 74 reg [31:0] trigger_level_dq36;
 75 reg [31:0] trigger_level_dq37;
 76 reg [31:0] trigger_level_dq38;
 77 reg [31:0] trigger_level_dq39;
 78 reg [31:0] trigger_level_dq40;
 79 reg [31:0] trigger_level_dq41;
 80 reg [31:0] trigger_level_dq42;
 81 reg [31:0] trigger_level_dq43;
 82 reg [31:0] trigger_level_dq44;
 83 reg [31:0] trigger_level_dq45;
 84 reg [31:0] trigger_level_dq46;
 85 reg [31:0] trigger_level_dq47;
 86 reg [31:0] trigger_level_dq48;
 87 reg [31:0] trigger_level_dq49;
 88 reg [31:0] trigger_level_dq50;
 89 reg [31:0] trigger_level_dq51;
 90 reg [31:0] trigger_level_dq52;
 91 reg [31:0] trigger_level_dq53;
 92 reg [31:0] trigger_level_dq54;
 93 reg [31:0] trigger_level_dq55;
 94 reg [31:0] trigger_level_dq56;
 95 reg [31:0] trigger_level_dq57;
 96 reg [31:0] trigger_level_dq58;
 97 reg [31:0] trigger_level_dq59;
 98 reg [31:0] trigger_level_dq60;
 99 reg [31:0] trigger_level_dq61;
 100 reg [31:0] trigger_level_dq62;
 101 reg [31:0] trigger_level_dq63;
 102 reg [31:0] trigger_level_dq64;

```

```
projects > risc-v > 20241209-V1.0_group5 > syn > dc.log | Warning 2 of 14 Aa ab * ↑ ↓ ≡ ×
1485 Note: Symbol # after min delay cost means estimated hold TNS across all active scenarios
1486 Optimization Complete
1487 -----
1488 Warning: Design 'full_chip' contains 2 high-fanout nets. A fanout number of 1000 will be used for delay calculations involving these
1489 nets. (TIM-134)
1490 Net 'u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_plic/u_sirv_plic_top/u_sirv_plic_main/
1491 flop_o_irq_plic_irq_o_dfflr/rst_n': 6388 load(s), 1 driver(s)
1492 Net 'u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/reset_nCatch_reg/reg_2/clk': 1279 load(s),
1493 driver(s)
1494 Writing ddc file 'results/full_chip_mapped_0.ddc'.
1495 Information: Starting from 2013.12 release, constant propagation is enabled even when boundary optimization is disabled. (OPT-1318)
1496 Information: Performing power optimization. (PWR-850)
1497 Alib files are up-to-date.
1498
1499 Information: There are 9833 potential problems in your design. Please run 'check_design' for more information. (LINT-99)
1500
```

## Warning TIM-134

根据实验 doc, 该 warning 可忽略

- Optimization 时的 warnings

```
projects > risc-v > 20241209-V1.0_group5 > syn > dclog | Warning 4 of 14 Aa ab * ↑ ↓ ≡ ×
3079 Note: Symbol # after min delay cost means estimated hold TNS across all active
3080 Optimization Complete
3081 -----
3082 Warning: Design 'full_chip' contains 2 high-fanout nets. A fanout number of 1000 will be used for delay calculations involving these
3083 nets. (TIM-134)
3084 Net 'u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_mems/u_expl_axi_icb2axi/u_sirv_gnrl_axi_buffer/
3085 o_axi_bresp_fifo/dp_gt0_vec_31_dfflr/rst_n': 6388 load(s), 1 driver(s)
3086 Net 'u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_cpu_top/u_e203_cpu/u_e203_core/u_e203_exu/u_e203_exu_csr/
3087 mcycleh_dfflr/clk_gate_qout_r_reg/CLK': 1279 load(s), 1 driver(s)
3088 Information: State dependent leakage is now switched from off to on.
3089 Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
3090 Warning: Design has unannotated black box outputs. (PWR-428)
3091 Writing ddc file 'results/full_chip_compile_ultra.ddc'.
3092 Warning: In the design e203_subsys_hclkgen, net 'hfextclk' is connecting multiple ports. (UCN-1)
3093 Warning: In the design sirv_jtaggpioport, net 'io_pins_TCK_i_ival' is connecting multiple ports. (UCN-1)
3094 Warning: In the design sirv_aon_wrapper, net 'io_pads_lfextclk_i_ival' is connecting multiple ports. (UCN-1)
3095 Warning: In the design e203_subsys_pll, net 'pllrefclk' is connecting multiple ports. (UCN-1)
3096 Warning: In the design e203_clk_ctrl, net 'clk' is connecting multiple ports. (UCN-1)
3097 Warning: In the design sirv_aon, net 'io_lfextclk' is connecting multiple ports. (UCN-1)
3098 Warning: In the design e203_reset_ctrl_MASTER1, net 'rst_core' is connecting multiple ports. (UCN-1)
3099 Writing ddc file 'results/full_chip_mapped.ddc'.
3100 Writing verilog file '/home/shikai/Soc-C-Assignment/projects/risc-v/20241209-V1.0_group5/syn/netlist/full_chip_mapped.v'.
3101 Warning: Verilog 'assign' or 'tran' statements are written out. (VO-4)
3102 Information: Annotated 'cell' delays are assumed to include load delay. (UID-282)
3103 Information: Writing timing information to file '/home/shikai/Soc-C-Assignment/projects/risc-v/20241209-V1.0_group5/syn/results/
full_chip_mapped.sdf'. (WT-3)
```

## 其余 Warning

- Warning TIM-134

- 根据实验 doc, 该 warning 可忽略

- Warning DvD-/I28

warning 可忽略  
g 'assign' or '

```
25 | remove_unconnec
26 | read_saif -inpu
27 | read_mif -inpu
28 | dc.tcl 由添加 Set
```

该命令对于数据输入

```

● shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/syn$ grep "assign" -c netlist/full_chip_mapped.v
10
● shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/syn$ grep "assign" -n netlist/full_chip_mapped.v
258: assign plloutclk = pllrefclk;
451: assign inspect_16m_clk = hfextclk;
477: assign rst_aon = rst_core;
478: assign rst_dtcm = rst_core;
479: assign rst_itcm = rst_core;
582: assign clk_aon = clk;
27415: assign inspect_core_clk = clk;
79833: assign io_jtag_TCK = io_pins_TCK_i_ival;
88159: assign io_lfclock = io_lfextclk;
91383: assign inspect_32k_clk = io_pads_lfextclk_i_ival;
○ shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/syn$ █

```

RTL 中存在的 10 处 assign 语句

因此要手动将 RTL 中这些 assign 赋值改为 CLKBUFUHDV4，从而消除网表中的 assign 语句：

- (1) assign plloutclk = pllout;

```

projects > risc-v > 20241209-V1.0_group5 > pre_sim > rtl > e203 > subsys > ┌ e203_subsys_pll.v
 48 31 module e203_subsys_pll(
 49 49 `endif//}
 50
 51 assign plloutclk = pllout;
 51+ // assign plloutclk = pllout;
 52+ CLKBUFUHDV4 u_buf_plloutclk(
 53+ .Z(plloutclk),
 54+ .I(pllout)
 55+);
 52 56 endmodule
 53 57

```

(1) 修改 e203\_subsys\_pll.v

- (2) assign inspect\_16m\_clk = hfextclk;

```

projects > risc-v > 20241209-V1.0_group5 > pre_sim > rtl > e203 > subsys > ┌ e203_subsys_hclkgen.v
 116 31 module e203_subsys_hclkgen(
 117 117 |
 118 assign inspect_16m_clk = hfextclk ;
 118+ // assign inspect_16m_clk = hfextclk ;
 119+ CLKBUFUHDV4 u_buf_inspect_16m_clk(
 120+ .Z(inspect_16m_clk),
 121+ .I(hfextclk)
 122+);
 119 123 assign inspect_pll_clk = plloutclk;
 120 124
 121 125 endmodule

```

(2) 修改 e203\_subsys\_hclkgen.v

- (3) assign rst\_aon = rst\_core;
- (4) assign rst\_dtcm = rst\_core;
- (5) assign rst\_itcm = rst\_core;

```
f E203_HAS_ITC
rst_itcm = rst
gn rst_itcm =
HDV4 u_buf_rst
st_itcm),
```

- ```
101      - assign rst_dtcm = rst_sync_n;
105+ // assign rst_dtcm = rst_sync_n;
106+ CLKBUFUHDV4 u_buf_rst_dtcm(
107+   .Z(rst_dtcm    ),
108+   .I(rst_sync_n  )
109+ );
110     `endif
111
104 112 // The Top always on clk and rst
105      - assign rst_aon = rst_sync_n;
113+ // assign rst_aon = rst_sync_n;
114+ CLKBUFUHDV4 u_buf_rst_aon(
115+   .Z(rst_aon    ),
116+   .I(rst_sync_n  )
117+ );
106 118
107 119 endmodule

(3) (4) (5) 修改 e203_reset_ctrl.v

• (6) assign clk_aon = clk;

projects > risc-v > 20241209-V1.0_group5 > pre_sim > rtl > e203 > core > e203_clk_ctrl.v
149  72 );
150 150 // The Top always on clk and rst
151      - assign clk_aon = clk;
151+ // assign clk_aon = clk;
152+ CLKBUFUHDV4 u_buf_clk_aon(
153+   .Z(clk_aon    ),
154+   .I(clk        )
155+ );
152 156
153 157 endmodule

(6) 修改 e203_clk_ctrl.v

• (7) assign inspect_core_clk = clk;

projects > risc-v > 20241209-V1.0_group5 > pre_sim > rtl > e203 > core > e203_cpu.v
891  32 )(
892 892 assign inspect_mem_rsp_ready = mem_icb_rsp_ready;
893      - assign inspect_core_clk = clk;
```

```
• (9) assign io_lfclk = io_lfextclk;

  projects > risc-v > 20241209-V1.0_group5 > pre_sim > rtl > e203 > perips > sirv_aon.v
    3600      78  );
    3601      3601  | assign io_in_0_e_ready = 1'h1;
    3602      -  assign io_lfclk = io_lfextclk;
    3602+     // assign io_lfclk = io_lfextclk;
    3603+     CLKBUFUHDV4 u_buf_io_lfclk(
    3604+       .Z(io_lfclk      ),
    3605+       .I(io_lfextclk )
    3606+     );
    3607
  3603  3607
```

```
• (10) assign inspect_32k_clk = io_pads_lfextclk_i_ival;  
  
projects > risc-v > 20241209-V1.0_group5 > pre_sim > rtl > e203 > perips > sirv_aon_wrapper.sv  
339   132 );  
340   340  
341     - assign inspect_32k_clk = aon_io_lfextclk;  
341+ // assign inspect_32k_clk = aon_io_lfextclk;  
342+ CLKBUFUHDV4 u_buf_inspect_32k_clk(  
343+   .Z(inspect_32k_clk ),  
344+   .I(aon_io_lfextclk )  
345+ );
```

(10) 修改 sirv_aon_wrapper.v

```
7      suppress_message VER-318
8      suppress_message ELAB-311
9      source -echo -verbose ./scripts/setup_dc.tcl
10     define_design_lib WORK -path ./WORK
11     analyze -format verilog ${RTL_SOURCE_VERILOG}
12     #analyze -format VHDL ${RTL_SOURCE_VHDL}
13     elaborate ${DESIGN_NAME}
14     write -hierarchy -format ddc -output ${RESULTS_DIR}/${DESIGN_NAME}.ddc
15     current_design ${DESIGN_NAME}
16     link
```

手动创建 WORK 工作目录

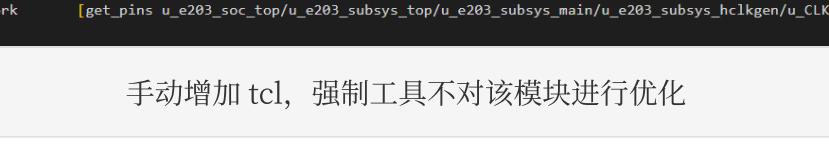
- [注意！！！]
修改 RTL 后进行重新综合，务必先退出 `dc_shell` 后再重新进入，然后加载 tcl。否则会报 Error!!!

```
看 full_chip_all_path_timing.rpt , 发现 CLKINUHDV3 例化后延迟超级大:

ects > risc-v > 20241209-V1.0_group5 > syn > reports > full_chip_all_path_timing.rpt
15 Wire Load Model Mode: top
16 Path Group: hextcik
17 Path Type: max
18
19 Point           Incr      Path
20 -----
21
22 clock hextcik (rise edge)      0.00    0.00
23 clock network delay (ideal)   6.00    6.00
24 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/reset_nCatch_reg/reg_0/q_reg/CK (DRQUHDV1)
25                                     0.00 # 6.00 r
26 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/reset_nCatch_reg/reg_0/q_reg/Q (DRQUHDV1)
27                                     0.55  6.55 r
28 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/reset_nCatch_reg/reg_0/q (sirv_AsyncResetReg_59)
29                                     0.00  6.55 r
30 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/reset_nCatch_reg/io_q[0] (sirv_AsyncResetRegVec_129_1)
31                                     0.00  6.55 r
32 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/U3/ZN (INUHDV1)
33                                     0.06  6.62 f
34 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/U5/ZN (MUX2NUHDV2)
35                                     0.16  6.77 r
36 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/U2/ZN (INUHDV1)
37                                     0.08  6.85 f
38 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_main_ResetCatchAndSync_2_1/io_sync_reset (sirv_ResetCatchAndSync_2_1)
39                                     0.00  6.85 f
```

```
76 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/U3/ZN (CLKINUHDV3)
77                                         1771.94 # 1778.79 r
78 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_clint/rst_n (e203_subsys_clint)
79                                         0.00 # 1778.79 r
80 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_clint/u_sirv_clint_top/rst_n (sirv_clint_top)
81                                         0.00 # 1778.79 r
82 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_clint/u_sirv_clint_top/U4/ZN (CLKINUHDV3)
83                                         1091.14 # 2869.93 f
84 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_clint/u_sirv_clint_top/u_sirv_clint/reset (sirv_clint)
85                                         0.00 2869.93 f
86 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_clint/u_sirv_clint_top/u_sirv_clint/time_1_reg_30/_RD (DRQUDHV0P7)
87                                         0.00 2869.93 f
88 data arrival time 2869.93
89
90 clock hfextclk (rise edge) 50.00 50.00
91 clock network delay (ideal) 6.00 56.00
92 clock uncertainty -0.50 55.50
93 u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_clint/u_sirv_clint_top/u_sirv_clint/time_1_reg_30/_CK (DRQUDHV0P7)
```

real_network，是为了暂时忽略复位网络的延时，认为其是理想的，看看其他路径是否有问题。因此，在 `cons.tcl` 中增加约束：



```
dc.log  cons.tcl M ×
Projects > risc-v > 20241209-V1.0_group5 > syn > scripts >  cons.tcl
9
0  #===== TIMING EXCEPTION =====#
1
2  set timing_non_uate_clock_compatibility  true
3  set enable_recovery_removal_arcs  true
4
5  set_ideal_network      [get_pins u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/U3/ZN]
6  # set_ideal_network      [get_pins u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_clint/u_sirv_clint_top/U4/ZN]
7  set_ideal_network      [get_pins u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_hclkgen/u_CLKMUX2/Z]
```

手动增加 tcl，强制工具不对该模块进行优化

```
projects > risc-v > 20241209-V1.0_group5 > syn > dc.log
3463 true
3464 set enable_recovery_removal_arcs true
3465 true
3466 set_ideal_network [get_pins u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/U3/ZN]
3467 Warning: Can't find object 'u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/U3/ZN' in design 'full_chip'. (UID-95)
3468 Error: Value for list 'object_list' must have 1 elements. (CMD-036)
3469
上面这个 tcl 找不到这个路径的模块

翻了一下 e203_subsys_main 的 RTL 代码，找不到 U3，U3是定义在 lib 里面的：
projects > lib > SMIC180 > SCC018UG_UHD_RVT_V0p4a > verilog > scc018ug_uhd_rvt.v
12861 else
```

```
12861     else
12862         `define SMC_NFORCE 1      // Flag to force output to x if notifer changes
12863     `endif
12864
12865     `celldefine
12866     module CLKINUHDV3 ( ZN, I);
12867     input I;
12868     output ZN;
12869 
```

被例化成 U3 的模块定义

```
矛报告 min_delay

-v > 20241209-V1.0.group5 > syn > reports > full_chip_constraint_all_violators.rpt
*****
min_delay/hold ('hfextclk' group)



| Endpoint                                                                                                                           | Required Path Delay | Actual Path Delay | Slack            |
|------------------------------------------------------------------------------------------------------------------------------------|---------------------|-------------------|------------------|
| u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_perips/u_perips_apb_pwm/u_tim0/u_in_stage/r_ls_clk_sync_reg_0/_D | 6.59                | 2.59 f            | -3.99 (VIOLATED) |
| u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_perips/u_perips_apb_pwm/u_tim1/u_in_stage/r_ls_clk_sync_reg_0/_D | 6.59                | 2.59 f            | -3.99 (VIOLATED) |
| u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_perips/u_perips_apb_pwm/u_tim2/u_in_stage/r_ls_clk_sync_reg_0/_D | 6.59                | 2.59 f            | -3.99 (VIOLATED) |
| u_e203_soc_top/u_e203_subsys_top/u_e203_subsys_main/u_e203_subsys_perips/u_perips_apb_pwm/u_tim3/u_in_stage/r_ls_clk_sync_reg_0/_D | 6.59                | 2.59 f            | -3.99 (VIOLATED) |



min_delay
```

```
93    set enable_recovery_removal_arcs          true
94
95    set_ideal_network           [get_pins u_e203_soc_top/u_e203_subsys_to_
96    # set_ideal_network           [get_pins u_e203_soc_top/u_e203_subsys_to_
97    set_ideal_network           [get_pins u_e203_soc_top/u_e203_subsys_to_
98
99    set_false_path -from lfextclk -to u_e203_soc_top/u_e203_subsys_to_
100   set_false_path -from lfextclk -to u_e203_soc_top/u_e203_subsys_to_
101   set_false_path -from lfextclk -to u_e203_soc_top/u_e203_subsys_to_
102   set_false_path -from lfextclk -to u_e203_soc_top/u_e203_subsys_to_
103
```

- 思考题五：为什么min_delay的-3.99的slack需要我们去关注？而max_delay的-1.3的slack则可以认为是正常的？
时序分析中 `min_delay` 的负 slack (-3.99) 需要我们特别关注，是由于它直接影响保持时间，它要确保数据在时钟边缘到达时能够在触发器输入端保持稳定。此类违规可能导致数据在采样时发生变化，从而引发逻辑错误和系统功能故障。日志中显示所有相关路径均存在严重的保持时间违规，表明这些路径的稳定性不足，必须优先解决。相较于，`max_delay` 的负 slack (-1.32) 虽然也显示出轻微的违规，但通常在高设计中会留有一定的冗余和裕量，因此这种小的负 slack 可以在不影响系统整体功能的情况下被接受，并可通过后续优化逐步改善。

情况下被接受，并可通过后续优化逐步改善。

```

1 set CLK_SKEW_SETUP 0.5
2 set CLK_SKEW_HOLD 0.5
3 set CLK_SOURCE_LATENCY 2
4 set CLK_NETWORK_LATENCY 4
5 set CLK_TRAN 1
6 set IN_DELAY_MASTER 3
7 set OUT_DELAY_MASTER 3
8 set IN_TRAN 0.5
9 set OUT_LOAD 15
10 set MAX_FANOUT 15
11 set MAX_CAP 0.25
12 set MAX_TRAN 2.4
13 reset_design
14 ===== CLK DEFINITION =====#
15 create_clock -period $CLK_HFEXTCLK_PERIOD -name hfextclk [get_ports hfextclk]
16 create_clock -period $CLK_LFEXTCLK_PERIOD -name lfextclk [get_ports lfextclk]
17 create_clock -period $CLK_JTAGCLK_PERIOD -name JTAG_CLK [get_ports io_pads_jtag_TCK_i_ival]

```

修改前的clock_uncertainty参数

```

1 set CLK_SKEW_SETUP 0.8
2 set CLK_SKEW_HOLD 0.7
3 set CLK_SOURCE_LATENCY 2
4 set CLK_NETWORK_LATENCY 4
5 set CLK_TRAN 1
6 set IN_DELAY_MASTER 3
7 set OUT_DELAY_MASTER 3
8 set IN_TRAN 0.5
9 set OUT_LOAD 15
10 set MAX_FANOUT 15
11 set MAX_CAP 0.25
12 set MAX_TRAN 2.4
13 reset_design
14 ===== CLK DEFINITION =====#
15 create_clock -period $CLK_HFEXTCLK_PERIOD -name hfextclk [get_ports hfextclk]
16 create_clock -period $CLK_LFEXTCLK_PERIOD -name lfextclk [get_ports lfextclk]
17 create_clock -period $CLK_JTAGCLK_PERIOD -name JTAG_CLK [get_ports io_pads_jtag_TCK_i_ival]

```

修改后的clock_uncertainty参数

● 面积报告

<pre> projects > risc-v > 20241209-V1.0_group5 > syn > reports > full_chip_mapped.area rpt 1 **** 2 **** 3 Report : area 4 Design : full_chip 5 Version: 0.2018.12.05.09 6 Date: 2024-01-19 22:48:17 2024 7 **** 8 Library(s) Used: 9 10 sc018ug_uhd_rvt_ss_vip62_125c_basic (file: /home/shikai/Soc-Assig 11 nment/projects/jlib/SMIC180/SP018_V1.5b/SP018_V1p5b/SP018_V1p5a 12 UMLIB (file: /home/shikai/Soc-Assigment/projects/jlib/SMIC180/SP 13 018_V1.5b/SP018_V1p5b/SP018_V1p5a) 14 SP018_V1p5a.lib (file: /home/shikai/Soc-Assigment/projects/jlib/SM 15 IC180/SP018_V1.5b/SP018_V1p5b/SP018_V1p5a) 16 Number of ports: 59727 17 Number of nets: 61483 18 Number of combinational cells: 47582 19 Number of sequential cells: 33974 20 Number of macrocells: 1509 21 Number of macro/black boxes: 88 22 Number of buf/inv: 9615 23 Number of references: 11 24 25 Combinational area: 361848.620496 26 Buf/Inv area: 61483.163072 27 Macro/Block Box area: 5163918.375800 28 Net Interconnect area: (No wire load specified) 29 30 Total cell area: 6067229.779885 31 Total area: undefined </pre>	<pre> projects > risc-v > 20241209-V1.0_group5 > syn > reports > full_chip_mapped.area rpt 1 **** 2 **** 3 Report : area 4 Design : full_chip 5 Version: 0.2018.12.05.09 6 Date: 2024-01-19 22:48:17 2024 7 **** 8 Library(s) Used: 9 10 sc018ug_uhd_rvt_ss_vip62_125c_basic (file: /home/shikai/Soc-Assig 11 nment/projects/jlib/SMIC180/SP018_V1.5b/SP018_V1p5b/SP018_V1p5a 12 UMLIB (file: /home/shikai/Soc-Assigment/projects/jlib/SMIC180/SP 13 018_V1.5b/SP018_V1p5b/SP018_V1p5a) 14 SP018_V1p5a.lib (file: /home/shikai/Soc-Assigment/projects/jlib/SM 15 IC180/SP018_V1.5b/SP018_V1p5b/SP018_V1p5a) 16 Number of ports: 58727 17 Number of nets: 96195 18 Number of cells: 47517 19 Number of combinational cells: 33989 20 Number of sequential cells: 15255 21 Number of macro/black boxes: 88 22 Number of buf/inv: 9617 23 Number of references: 11 24 25 Combinational area: 361965.279699 26 Buf/Inv area: 61369.594372 27 Macro/Block Box area: 5163918.375800 28 Net Interconnect area: (No wire load specified) 29 30 Total cell area: 6067229.399088 31 Total area: undefined </pre>
修改前\	修改后\

● 时序报告

<pre> Point Fanout Trans Incr Path Attributes -----+-----+-----+-----+-----+-----+ clock_burstk(cris edge) 956.64 clock network delay (delay) 956.64 clock uncertainty (delay) 956.64 clock uncertainty (max) 956.64 clock uncertainty (min) 956.64 clock uncertainty (slack) 956.64 clock uncertainty (total) 956.64 clock uncertainty (worst) 956.64 clock setup time (delay) 956.64 clock setup time (max) 956.64 clock setup time (min) 956.64 clock setup time (slack) 956.64 clock setup time (total) 956.64 clock setup time (worst) 956.64 data arrival time (delay) 956.64 data arrival time (max) 956.64 data arrival time (min) 956.64 data arrival time (slack) 956.64 data arrival time (total) 956.64 data arrival time (worst) 956.64 data required time (delay) 956.64 data required time (max) 956.64 data required time (min) 956.64 data required time (slack) 956.64 data required time (total) 956.64 data required time (worst) 956.64 slack (NET) 48.73 </pre>	<pre> Point Fanout Trans Incr Path Attributes -----+-----+-----+-----+-----+-----+ clock_burstk(cris edge) 956.64 clock network delay (delay) 956.64 clock uncertainty (delay) 956.64 clock uncertainty (max) 956.64 clock uncertainty (min) 956.64 clock uncertainty (slack) 956.64 clock uncertainty (total) 956.64 clock setup time (delay) 956.64 clock setup time (max) 956.64 clock setup time (min) 956.64 clock setup time (slack) 956.64 clock setup time (total) 956.64 clock setup time (worst) 956.64 data arrival time (delay) 956.64 data arrival time (max) 956.64 data arrival time (min) 956.64 data arrival time (slack) 956.64 data arrival time (total) 956.64 data arrival time (worst) 956.64 data required time (delay) 956.64 data required time (max) 956.64 data required time (min) 956.64 data required time (slack) 956.64 data required time (total) 956.64 data required time (worst) 956.64 slack (NET) 48.43 </pre>
修改前\	修改后\

I. 面积分析

- 约束参数变化
 - 时钟偏斜 (CLK_SKEW_SETUP 和 CLK_SKEW_HOLD): 这两个参数从0.5和0.5增加到0.8和0.7, 时钟偏斜范围增大。这可能会导致时序收敛情况的变化。
 - 其他参数: 如 IN_DELAY_MASTER 和 OUT_DELAY_MASTER 保持不变。
- 综合结果变化
 - 组合面积:
 - 原始报告: 361848.620496
 - 调整后报告: 361965.279699
 - 变化: 增加了约116.659203。
 - 缓冲区/反相器面积:
 - 原始报告: 61483.163072
 - 调整后报告: 61509.505472
 - 变化: 增加了约26.3424。
 - 非组合面积和宏/黑盒面积保持不变。
 - 总 Cell 面积:
 - 原始报告: 6067112.739885
 - 调整后报告: 6067229.399088
 - 变化: 增加了约116.659203。
- 变化原因分析
 - 时钟偏斜增加: 增加时钟偏斜可能导致综合工具在时序收敛时选择更多的缓冲器和逻辑单元, 以满足新的时序要求, 从而导致面积的增加, 因为更多的逻辑单元和缓冲器被引入。
 - 时序收敛: 在面对较大的时钟偏斜时, 需要更多的逻辑来确保时序的稳定性, 从而导致组合面积和缓冲区面积的增加。

II. 时序分析

- 1. 时序路径分析
 - 时钟 hfextclk :
 - 数据到达时间: 956.64
 - 数据要求时间: 956.64
 - 时序裕量 (slack): 48.73 (修改前), 48.43 (修改后)

从时序报告中可以看到, hfextclk 的到达时间和要求时间没有变化, 因此时序裕量的变化主要是由于其他因素的影响。
 - 时钟 JTAG_CLK :
 - 数据到达时间: 1005.37 (修改前), 1005.20 (修改后)
 - 数据要求时间: 1005.50 (修改前), 1005.20 (修改后)
 - 时序裕量 (slack): -0.13 (修改前), -0.13 (修改后)

JTAG_CLK 的时序裕量保持不变, 尽管数据到达时间有所减少, 但由于要求时间也减少, 因此裕量没有变化。
- 2. 变化原因分析
 - 时序裕量的变化:
 - hfextclk 的 slack 从 48.73 减少到 48.43, 虽然变化不大, 但表明在新的约束下, 时序的稳定性略有下降。这可能是由于时钟偏斜的增加导致时序路径的延迟略微增加。
 - JTAG_CLK时序裕量保持不变:
 - 尽管数据到达时间有所减少, 但要求时间的减少使得 slack 保持不变。这表明在新的约束下, JTAG_CLK 的时序路径得到了更好的优化, 或者设计的复杂度没有显著增加。
- 3. 结果综合影响
 - 时序的稳定性: 尽管整体时序裕量仍然是正值, 但 hfextclk 的 slack 略有下降, 表明在新的设计约束下, 时序稳定性受到了一定影响。

LAB10: 逻辑综合4: 提交网表和约束

- 数据交付后端

```

shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/syn_bak$ find . -name full_chip_mapped.v
./netlist/full_chip_mapped.v
shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/syn_bak$ find . -name full_chip_mapped.sdc

```

逻辑综合完成后的full_chip_mapped结果文件

- 备份 ./syn --> ./syn_bak

```

shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/syn_bak$ tree -L 2 .
+-- alib-52
|   +-- sc018ug_uhd_rvt_ss_vip62_125c_basic.db.alib
|   +-- command_log
|   +-- config.v
|   +-- default.svf
|   +-- filenames.log
|   +-- i2c_masterDefines.v
|   +-- netlist
|       +-- full_chip_mapped.v
|           +-- README.md
|       +-- netlist
|           +-- full_chip_mapped
|               +-- README.md
|       +-- reports
|           +-- full_chip_all_path_timing.rpt
|           +-- full_chip_check_timing_final.rpt
|           +-- full_chip_constraint_all_violators.rpt
|           +-- full_chip_mapped.area.rpt
|           +-- full_chip_mapped.cell.rpt
|           +-- full_chip_mapped.clock_gating.rpt
|           +-- full_chip_mapped.power.rpt
|           +-- full_chip_mapped.timing.rpt
|           +-- full_chip_report_hierarchy.rpt
|           +-- full_chip_report_timing_requirements.rpt
|               +-- README.md
|       +-- results
|           +-- full_chip_compile_ultra.ddc
|           +-- full_chip_mapped.0.ddc
|           +-- full_chip_mapped.ddc
|           +-- full_chip_mapped.sdc
|           +-- full_chip_mapped.sdf
|           +-- full_chip_read.ddc
|               +-- README.md
|           +-- scripts
|               +-- clean.sh
|               +-- config.tcl
|               +-- dft.tcl
|               +-- netlist.tcl
|                   +-- ADV_TIMER_APB_IF.mv
|                   +-- adv_timer_apb_if-verilog.pvl
|                   +-- adv_timer_apb_if-verilog.syn
|               +-- WORK
|                   +-- ADV_TIMER_APB_IF.mv
|                   +-- adv_timer_apb_if-verilog.pvl
|                   +-- adv_timer_apb_if-verilog.syn

```

备份逻辑综合目录到syn_bak

LAB11: 后仿真 post_sim

- 建立后仿环境

```

shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/post_sim$ tree -L 2 .
+-- clean.sh
+-- config.v
+-- e203_defines.v
+-- e203_filelist.f
+-- e203.fsd
+-- riscv-tools
|   +-- build.common
|   +-- build.e203-rvtests.sh
|   +-- fpga_test4sim
|   +-- README.md
|   +-- riscv-tests
+-- run_vcs.csh
+-- run_verdi.csh
+-- tb
    +-- tb_top.v

```

后仿真的工作目录

- 修改 filelist

```

shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/post_sim$ run_vcs.csh

```

删除 RTL 部分的 filelist, 添加 full_chip_mapped.v

- 修改测试激励

```

shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/post_sim$ tree -L 2 .
+-- module tb_top();
+-- define PC_THR IRQ_BEFOR_MRET E203_PC_SIZE'h800000d6
+-- define PC_AFTER_SETMVEC E203_PC_SIZE'h80000015c
+-- assign rf_r_3_ = {
+--     rf_r_3_31_ = 'EXU.u.e203_exu.regfile.rf_r_3_31_;
+--     rf_r_3_30_ = 'EXU.u.e203_exu.regfile.rf_r_3_30_;
+--     rf_r_3_29_ = 'EXU.u.e203_exu.regfile.rf_r_3_29_;
+--     rf_r_3_28_ = 'EXU.u.e203_exu.regfile.rf_r_3_28_;
+--     rf_r_3_27_ = 'EXU.u.e203_exu.regfile.rf_r_3_27_;
+--     rf_r_3_26_ = 'EXU.u.e203_exu.regfile.rf_r_3_26_;
+--     rf_r_3_25_ = 'EXU.u.e203_exu.regfile.rf_r_3_25_;
+--     rf_r_3_24_ = 'EXU.u.e203_exu.regfile.rf_r_3_24_;
+--     rf_r_3_23_ = 'EXU.u.e203_exu.regfile.rf_r_3_23_;
+--     rf_r_3_22_ = 'EXU.u.e203_exu.regfile.rf_r_3_22_;
+--     rf_r_3_21_ = 'EXU.u.e203_exu.regfile.rf_r_3_21_;
+--     rf_r_3_20_ = 'EXU.u.e203_exu.regfile.rf_r_3_20_;
+--     rf_r_3_19_ = 'EXU.u.e203_exu.regfile.rf_r_3_19_;
+--     rf_r_3_18_ = 'EXU.u.e203_exu.regfile.rf_r_3_18_;
+--     rf_r_3_17_ = 'EXU.u.e203_exu.regfile.rf_r_3_17_;
+--     rf_r_3_16_ = 'EXU.u.e203_exu.regfile.rf_r_3_16_;
+--     rf_r_3_15_ = 'EXU.u.e203_exu.regfile.rf_r_3_15_;
+--     rf_r_3_14_ = 'EXU.u.e203_exu.regfile.rf_r_3_14_;
+--     rf_r_3_13_ = 'EXU.u.e203_exu.regfile.rf_r_3_13_;
+--     rf_r_3_12_ = 'EXU.u.e203_exu.regfile.rf_r_3_12_;
+--     rf_r_3_11_ = 'EXU.u.e203_exu.regfile.rf_r_3_11_;
+--     rf_r_3_10_ = 'EXU.u.e203_exu.regfile.rf_r_3_10_;
+--     rf_r_3_9_ = 'EXU.u.e203_exu.regfile.rf_r_3_9_;
+--     rf_r_3_8_ = 'EXU.u.e203_exu.regfile.rf_r_3_8_;
+--     rf_r_3_7_ = 'EXU.u.e203_exu.regfile.rf_r_3_7_;
+--     rf_r_3_6_ = 'EXU.u.e203_exu.regfile.rf_r_3_6_;
+--     rf_r_3_5_ = 'EXU.u.e203_exu.regfile.rf_r_3_5_;
+--     rf_r_3_4_ = 'EXU.u.e203_exu.regfile.rf_r_3_4_;
+--     rf_r_3_3_ = 'EXU.u.e203_exu.regfile.rf_r_3_3_;
+--     rf_r_3_2_ = 'EXU.u.e203_exu.regfile.rf_r_3_2_;
+--     rf_r_3_1_ = 'EXU.u.e203_exu.regfile.rf_r_3_1_;
+--     rf_r_3_0_ = 'EXU.u.e203_exu.regfile.rf_r_3_0_;

```

修改测试激励 tb_top.v 中 x3 的赋值 (从 full_chip_mapped 中取值)

```

shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/post_sim$ tree -L 2 .
+-- clean.sh
+-- config.v
+-- e203_defines.v
+-- e203_filelist.f
+-- e203.fsd
+-- riscv-tools
|   +-- build.common
|   +-- build.e203-rvtests.sh
|   +-- fpga_test4sim
|   +-- README.md
|   +-- riscv-tests
+-- run_vcs.csh
+-- run_verdi.csh
+-- tb
    +-- tb_top.v

```

后仿真的工作目录

- 修改 filelist

```

shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/post_sim$ run_vcs.csh

```

删除 RTL 部分的 filelist, 添加 full_chip_mapped.v

- 修改测试激励

```

shikai@ubuntu:~/SoC-Assignment/projects/risc-v/20241209-V1.0_group5/post_sim$ tree -L 2 .
+-- module tb_top();
+-- define PC_THR IRQ_BEFOR_MRET E203_PC_SIZE'h800000d6
+-- define PC_AFTER_SETMVEC E203_PC_SIZE'h80000015c
+-- assign rf_r_3_ = {
+--     rf_r_3_31_ = 'EXU.u.e203_exu.regfile.rf_r_3_31_;
+--     rf_r_3_30_ = 'EXU.u.e203_exu.regfile.rf_r_3_30_;
+--     rf_r_3_29_ = 'EXU.u.e203_exu.regfile.rf_r_3_29_;
+--     rf_r_3_28_ = 'EXU.u.e203_exu.regfile.rf_r_3_28_;
+--     rf_r_3_27_ = 'EXU.u.e203_exu.regfile.rf_r_3_27_;
+--     rf_r_3_26_ = 'EXU.u.e203_exu.regfile.rf_r_3_26_;
+--     rf_r_3_25_ = 'EXU.u.e203_exu.regfile.rf_r_3_25_;
+--     rf_r_3_24_ = 'EXU.u.e203_exu.regfile.rf_r_3_24_;
+--     rf_r_3_23_ = 'EXU.u.e203_exu.regfile.rf_r_3_23_;
+--     rf_r_3_22_ = 'EXU.u.e203_exu.regfile.rf_r_3_22_;
+--     rf_r_3_21_ = 'EXU.u.e203_exu.regfile.rf_r_3_21_;
+--     rf_r_3_20_ = 'EXU.u.e203_exu.regfile.rf_r_3_20_;
+--     rf_r_3_19_ = 'EXU.u.e203_exu.regfile.rf_r_3_19_;
+--     rf_r_3_18_ = 'EXU.u.e203_exu.regfile.rf_r_3_18_;
+--     rf_r_3_17_ = 'EXU.u.e203_exu.regfile.rf_r_3_17_;
+--     rf_r_3_16_ = 'EXU.u.e203_exu.regfile.rf_r_3_16_;
+--     rf_r_3_15_ = 'EXU.u.e203_exu.regfile.rf_r_3_15_;
+--     rf_r_3_14_ = 'EXU.u.e203_exu.regfile.rf_r_3_14_;
+--     rf_r_3_13_ = 'EXU.u.e203_exu.regfile.rf_r_3_13_;
+--     rf_r_3_12_ = 'EXU.u.e203_exu.regfile.rf_r_3_12_;
+--     rf_r_3_11_ = 'EXU.u.e203_exu.regfile.rf_r_3_11_;
+--     rf_r_3_10_ = 'EXU.u.e203_exu.regfile.rf_r_3_10_;
+--     rf_r_3_9_ = 'EXU.u.e
```