实验二　简单处理器（CPU）的逻辑综合实验

一、实验要求

1. 利用 VCS 和已有的 test 文件，对 CPU 进行测试，验证 RTL 级 CPU 的正确性，并解决验证过程中遇到的问题。

注：在 cpu 工作目录下使用 make 命令对 CPU 进行验证，并根据提示执行相应的命令运行 test 文件对 CPU 进行测试。

输入“ucli%>call test(1);run”运行 CPUtest1.dat 测试文件。

运行以上文件，应显示如下结果：

```
ucli% call test(1);run
RUNNING THE BASIC DIAGOSTIC TEST
THIS TEST SHOULD HALT WITH PC = 17
PC INSTR OP DATA ADR
-- ----- -- ---- ---
*Verdi* Loading libsscore_vcs202206.so
FSDB Dumper for VCS, Release Verdi_T-2022.06, Linux x86_64/64bit, 05/29/2022
(C) 1996 - 2022 by Synopsys, Inc.
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the programs that are using this file.
*Verdi* : Create FSDB file 'cpu_test.fsdb'
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : Enable +all dumping.
*Verdi* : End of traversing.
xx HLT   0  zz  zz  00
00 JMP   7  zz  fe  00
1e JMP   7  zz  e3  1e
03 LDA   5  zz  ba  03
04 SKZ   1  zz  20  04
06 LDA   5  zz  bb  06
07 SKZ   1  zz  20  07
08 JMP   7  zz  ea  08
0a STO   6  zz  dc  0a
0b LDA   5  zz  ba  0b
0c STO   6  zz  dc  0c
0d LDA   5  zz  bc  0d
0e SKZ   1  zz  20  0e
10 XOR   4  zz  9b  10
11 SKZ   1  zz  20  11
12 JMP   7  zz  f4  12
14 XOR   4  zz  9b  14
15 SKZ   1  zz  20  15
17 HLT   0  zz  00  17
HALTED AT PC = 17

************************************************************
*        THE FOLLOWING DEBUG TASKS ARE AVAILABLE:          *
* Enter "call test(1);run" to run the 1st diagnostic program.   *
* Enter "call test(2);run" to run the 2nd diagnostic program.   *
* Enter "call test(3);run" to run the Fibonacci program.        *
* Enter "call test(4);run" to run the COUNTER program.         *
* Enter "call test(5);run" to run the 2^n program.        *
************************************************************

cpu_test.v, 69 :       $stop ;
```

输入“ucli%>call test(2);run”运行 CPUtest2.dat 测试文件。

运行以上文件，应显示如下结果：

```
ucli% call test(2);run
RUNNING THE ADVANCED DIAGOSTIC TEST
THIS TEST SHOULD HALT WITH PC = 10
PC INSTR OP DATA ADR
-- ----- -- ---- ---
00 LDA   5  zz  bb  00
01 AND   3  zz  7c  01
02 XOR   4  zz  9b  02
03 SKZ   1  zz  20  03
05 ADD   2  zz  5a  05
06 SKZ   1  zz  20  06
07 JMP   7  zz  e9  07
09 XOR   4  zz  9c  09
0a ADD   2  zz  5a  0a
0b STO   6  zz  dd  0b
0c LDA   5  zz  ba  0c
0d ADD   2  zz  5d  0d
0e SKZ   1  zz  20  0e
10 HLT   0  zz  00  10
HALTED AT PC = 10

*********************************************************
*        THE FOLLOWING DEBUG TASKS ARE AVAILABLE:       *
* Enter "call test(1);run" to run the 1st diagnostic program.   *
* Enter "call test(2);run" to run the 2nd diagnostic program.   *
* Enter "call test(3);run" to run the Fibonacci program.        *
* Enter "call test(4);run" to run the COUNTER program.        *
* Enter "call test(5);run" to run the 2^n program.       *
*********************************************************

cpu_test.v, 69 :       $stop ;
```

输入“ucli%>call test(3);run”运行 CPUtest3.dat 测试文件。

运行以上文件，应显示如下结果：

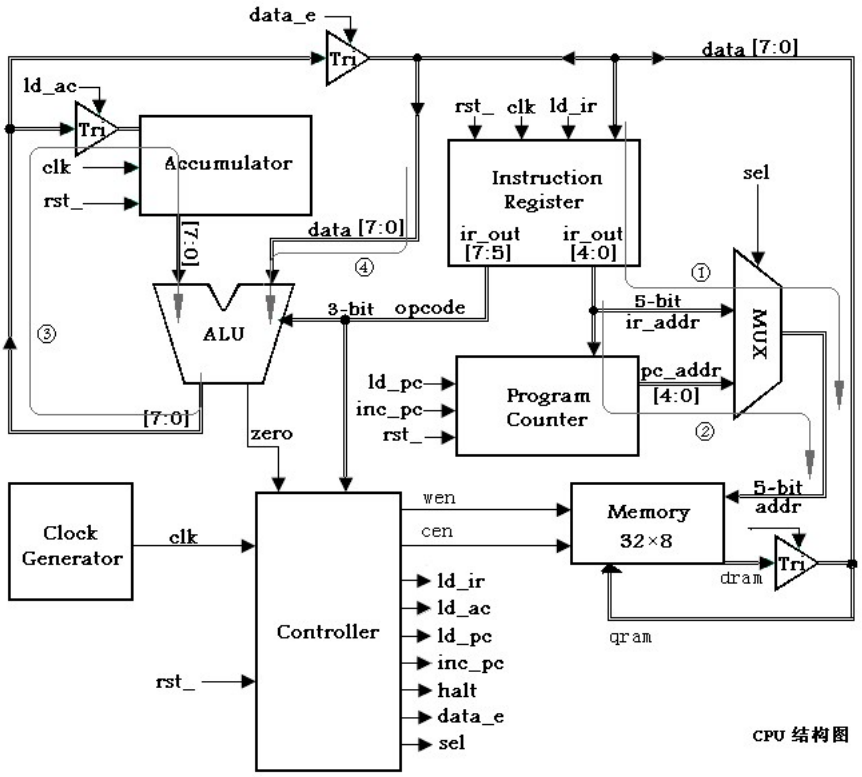
```
ucli% call test(3);run
RUNNING THE FIBONACCI CALCULATOR
THIS PROGRAM SHOULD CALCULATE TO 144
FIBONACCI NUMBER
 ---------------
  0
  1
  1
  2
  3
  5
  8
 13
 21
 34
 55
 89
144
HALTED AT PC = 0c

*********************************************************
*        THE FOLLOWING DEBUG TASKS ARE AVAILABLE:       *
* Enter "call test(1);run" to run the 1st diagnostic program.   *
* Enter "call test(2);run" to run the 2nd diagnostic program.   *
* Enter "call test(3);run" to run the Fibonacci program.        *
* Enter "call test(4);run" to run the COUNTER program.        *
* Enter "call test(5);run" to run the 2^n program.       *
*********************************************************

cpu_test.v, 69 :       $stop ;
```

2. 对 CPU 添加输入输出 PAD，编写脚本，利用 EDA 逻辑工具和工艺库文件对生成的顶层文件进行逻辑综合，生成门级网表和 SDF 文件。分别设置时钟频率 150MHz、50MHz、20MHz，分析逻辑综合结果是否满足时序要求。

3. 利用之前的 test 文件，反标 SDF 后，对逻辑综合生成的门级网表（150MHz、50MHz、20MHz）进行功能验证，并对门级验证的结果进行分析。



二、注意事项

1. 逻辑综合采用的工艺库所在的路径和库名：/home/lib
2. 逻辑综合用的 CPU 文件的目录：/home/cpu
3. 逻辑综合的一个参考示例见附录（注意该示例与实验要求不同!!!）。

附录：

（1）建立综合目录 syn，如图 2.1 所示，建立命令为：

mkdir syn

图 2.1 建立工作目录 syn

在 syn 文件夹下建立 ref、rpt、rtl、scripts、unmapped、mapped 等文件夹，如图 2.2 所示。

图 2.2 建立目录

（2）设置工作环境

首先复制综合要使用的单元库和符号库到 ref 目录里。

在工作目录 syn 下，用 vi 编辑器编写设置脚本文件 common_setup.tcl、dc_setup.tcl 和.synopsys_dc.setup(注意前面有个点)。在工作目录下启动 Design Compiler(DC)时，会自动运行.synopsys_dc.setup 文件，对 DC 的运行环境进行配置，并指定综合所需要的工艺库。参考.synopsys_dc.setup 脚本文件如下，请确认单元库的绝对路径和下面脚本中的一致。

```
#.synopsys_dc.setup
history keep 290
alias h history
alias rc "report_constraint -all_violators"
alias rt report _timing
alias ra report_area
alias rq report_qor
alias page_on {set sh_enable_page_mode true}
alias page_off (set sh_enable_page_mode_false)
alias fr "remove design -designs”
source common setup.tcl
source dc setup.tcl
```

.synopsys_dc.setup 运行时调用文件 common_setup.tcl、dc_setup.tcl，分别如下：

```
#common_setup.tcl
Set ADDITIONAL _SEARCH PATH
"/home/autumn/ASIC/syn ./unmapped ./rtl ./scripts" ;#Directories containing logical libraries,
set TARGET LIBRARY FILES
"/home/autumn/ASIC/syn/ref/typical_1v2c25.db
/home/autumn/ASIC/syn/ref/SP013D3V1p2_typ.db";
set SYMBOL LIBRARY FILES
#Symbol library fiTe
"/home/autumn/ASIC/syn/ref/smic13g.sdb";
```

（3）添加 PAD

编写 control_pad.v 作为顶层文件，从库文件中选择并实例化合适的 IO 单元。

```
timescale 1ns/1ns
module control_pad
(
input wire rst_
input wire clk,
input wire zero,
input wire[2:0]opcode,
output wire rd,
output wire wr,
output wire ld_ir,
output wire ld_ac,
output wire ld_pc,
```

```
output wire inc_pc,
output wire halt,
output wire data_e,
output wire sel
);
wire rd_pad;
wire wr_pad;
wire ld_ir_pad;
wire ld_ac_pad;
wire ld_pc_pad;
wire inc_pc_pad;
wire halt_pad;
wire data_e_pad;
wire sel_pad;
wire[2:0]opcode_pad;
wire zero_pad;
wire clk_pad;
wire rst_pad;
PI i_rst(.PAD(rst_),.C(rst_pad));
PI i_clk(.PAD(clk),.C(clk_pad));
PI i_zero(.PAD(zero),.C(zero_pad));
PI i_opcode_0(.PAD(opcode[0]),.C(opcode_pad[0]));
PI i_opcode_1(.PAD(opcode[1]),.C(opcode_pad[1]));
PI i_opcode_2(.PAD(opcode[2]),.C(opcode_pad[2]));
PO8 i_rd(.I(rd_pad),.PAD(rd));
PO8 i_wr(.I(wr_pad),.PAD(wr));
PO8 i_ld_ir(.I(ld_ir_pad),.PAD(ld_ir));
Po8 i_ld_ac(.I(ld_ac_pad),.PAD(1d_ac));
PO8 i_ld_pc(.I(ld_pc_pad),.PAD(1d_pc));
PO8 i_inc_pc(.I(inc_pc_pad),.PAD(inc_pc));
PO8 i_halt(.I(halt_pad),.PAD(halt));
PO8 i_data_e(.I(data_e_pad),.PAD(data_e));
PO8 i_sel(.I(sel_pad),.PAD(sel));
control i_control

(
.rst_(rst pad)
.clk(clk_pad),
.rd(rd_pad),
.wr(wr_pad),
.ld_ir(ld_ir_pad),
.ld_ac(ld_ac_pad),
.ld_pc(ld_pc_pad),
.inc_pc(inc_pc_pad),
```

```
    .halt(halt_pad),
    .data_e(data_e_pad),
    .sel(sel_pad),
    .opcode(opcode_pad),
    .zero(zero_pad)
    );
    endmodule
```

（4）编写综合脚本

在 scripts 目录下用 vi 编辑器编写脚本文件 dc_scripts.tcl。在编写脚本文件时注意和要综合的设计名称及端口的一致性。参考脚本文件如下。

```
#dc_scripts.tcl
read_file -format verilog ./rtl/control.v
read_file -format verilog ./rtl/control_pad.v
write -hierarchy -f ddc -out unmapped/control_pad.ddc
list designs
list_libs
set lib_name typical_1v2c25
current_design control_pad
link
write -hierarchy -f ddc -out unmapped/control_pad.ddc
list designs
list libs
#Create clock object and set uncertainty
create_clock -period 20 [get_ports clk]
set_clock_uncertainty 0.2 [get_clocks clk]
#Set constraints on input ports
suppress_message UID-401
set_driving_cell -library slib_name -lib_cell AND2X4[remove_from_collection
[all_inputs][get_ports clk]]
set_input_delay 0.1 -max -clock clk [remove_from_collection [all_inputs][get_ports clk]]
#set_input_delay 1.2 -max -clock clock [get_ports Neg_Flag]
#Set constraints on output ports
set_output_delay 1 -max -clock clk [all_outputs]
set_load [expr [load_of slib_name/AND2X4/A]*15][all_outputs]
set_dont_touch i_rst true
set_dont_touch i_clk true
set_dont_touch i_zero true
set_dont_touch i_opcode_0 true
set_dont_touch i_opcode_1 true
set_dont_touch i_opcode_2 true
set_dont_touch i_sel true
set_dont_touch i_data_e true
```

```
set_dont_touch i_inc_pc true
set_dont_touch i_ld_pc true
set_dont_touch i_ld_ac true
set_dont_touch i_ld_ir true
set_dont_touch i_wr true
set_dont_touch i_rd true
set_dont_touch i_halt true
set_dont_touch_network opcode[0]
set_dont_touch_network opcode[1]
set_dont_touch_network opcode[2]
compile_ultra
report_constraint -all > ./rpt/rpt_consitraints
report_timing > ./rpt/rpt_timing
report_area > ./rpt/rpt_area
report_power > ./rpt/rpt_power
write -hierarchy -format ddc -output ./mapped/control_pad.ddc
write -hierarchy -format verilog -output ./mapped/control_pad.v
write_sdc ./mapped/control_pad.sdc
write_sdf ./mapped/control_pad.sdf
list_designs
list_libs
```

（5）综合的执行

在工作目录下使用 dc_shell-64 命令启动 Design Compiler,输入以下命令并回车：

source ./scripts/dc_scripts.tcl

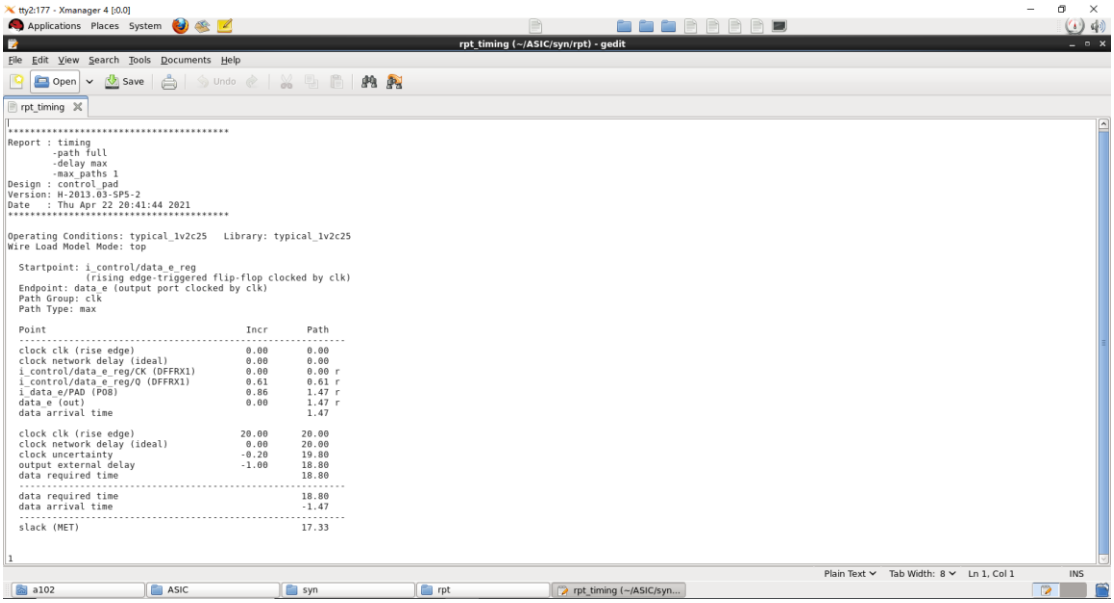
该命令运行/scripts/dc_scripts.tcl 文件并对设计进行综合，能实时观察综合结果。综合成功后，输出网表、时序约束文件和各种分析报告，其中，网表和时序约束文件保存在./mapped 目录下，各种分析报告保存在./rpt 目录下。

在综合过程中，注意查看报告的 ERROR 和 Warning,及时对引起问题的错误进行修改。

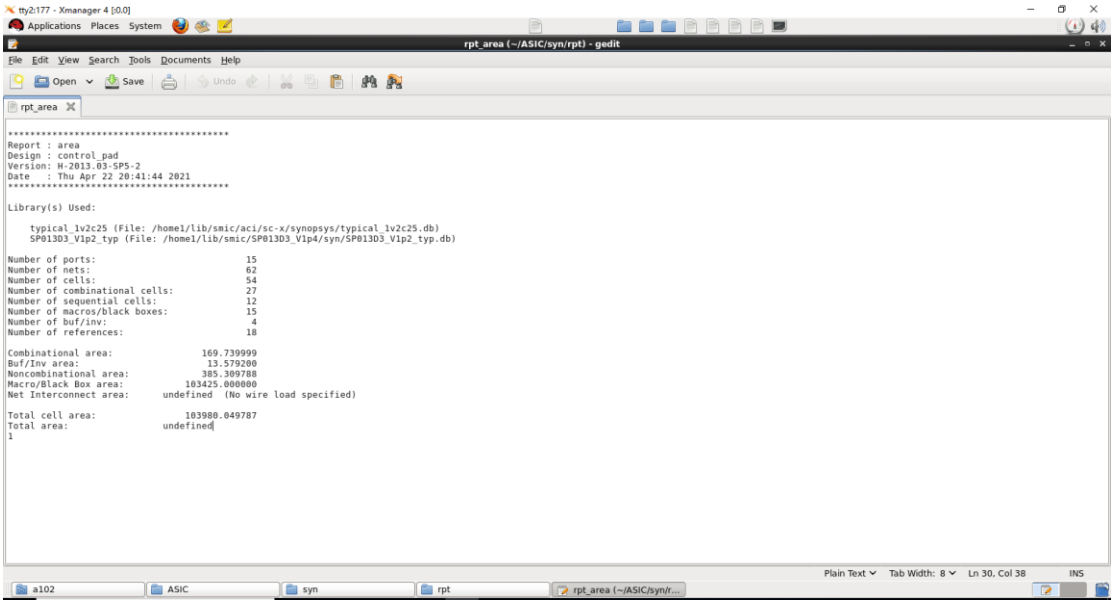
（6）综合结果分析

在目录 rpt 和 mapped 下查看各种输出结果和分析报告，修改./script/dc_script.tcl 中时钟约束和面积约束，查看其对综合结果的影响。以下图片分别是时序分析报告、面积分析报告、功耗分析报告、综合网表以及时序约束文件。

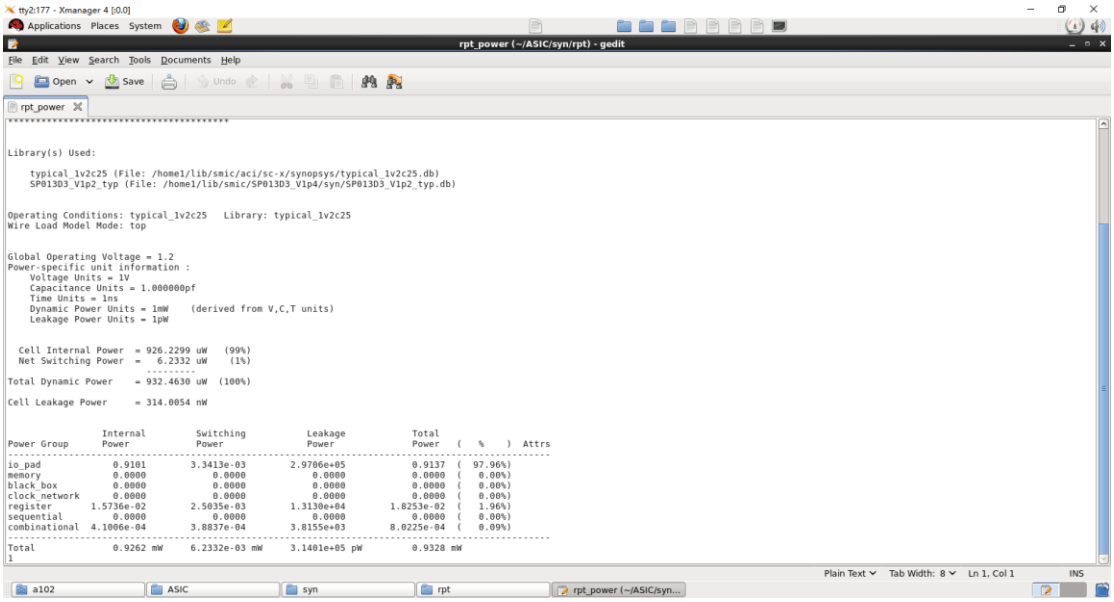
时序分析报告：



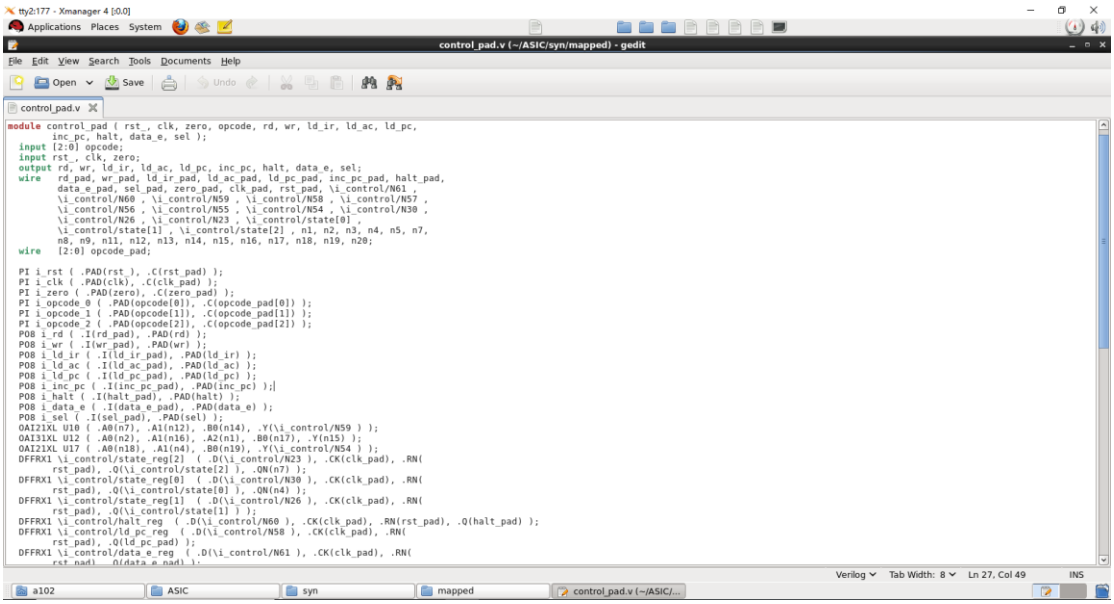
面积分析报告：



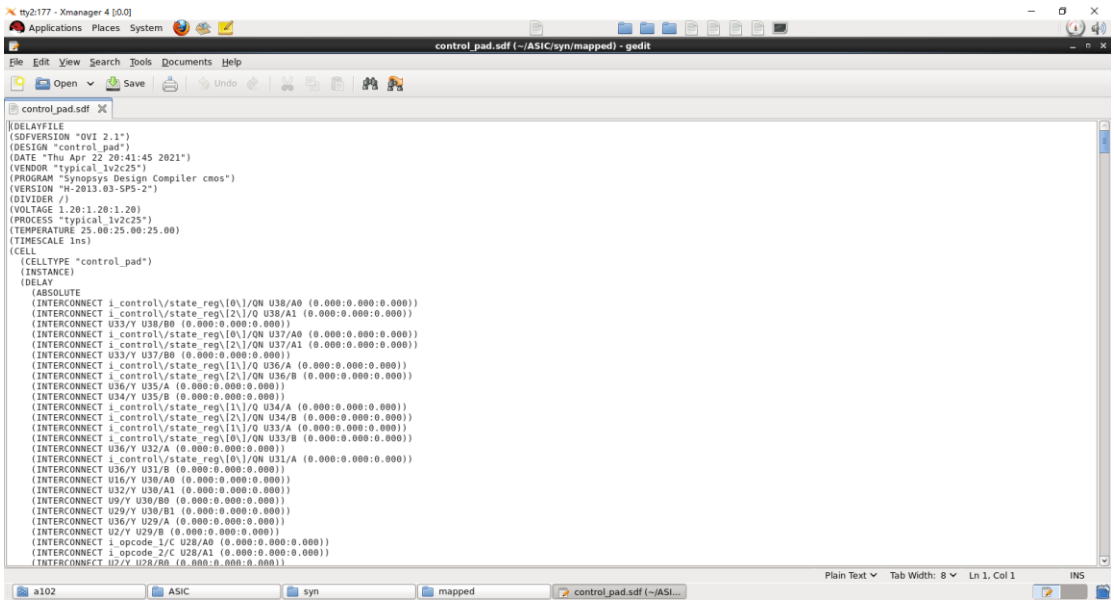
功耗分析报告：

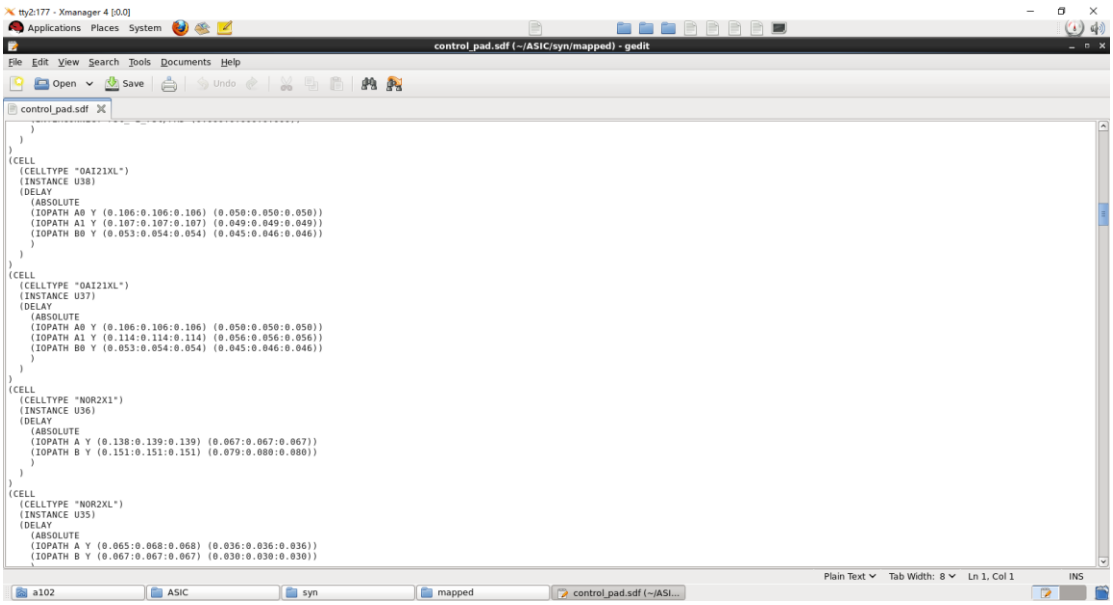


综合生成的网表：



综合生成的时序约束文件：


tty2:177 - Xmanager 4 [0.0]
Applications Places System
control_pad.sdf (~/ASIC/syn/mapped) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
control_pad.sdf
(DELAYFILE
(SDFVERSION "OVI 2.1")
(DESIGN "control_pad")
(DATE "Thu Apr 22 20:41:45 2021")
(VENDOR "typical_1v2c25")
(PROGRAM "Synopsys Design Compiler cmos")
(VERSION "H-2013.03-SP5-2")
(DIVIDER /)
(VOLTAGE 1.20:1.20:1.20)
(PROCESS "typical_1v2c25")
(TEMPERATURE 25.00:25.00:25.00)
(TIMESCALE 1ns)
(CELL
(CELLTYPE "control_pad")
(INSTANCE)
(DELAY
(ABSOLUTE
(INTERCONNECT i_control\/state_reg\[0\]/QN U38/A0 (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[2\]/Q U38/A1 (0.000:0.000:0.000))
(INTERCONNECT U33/Y U38/B0 (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[0\]/QN U37/A0 (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[2\]/QN U37/A1 (0.000:0.000:0.000))
(INTERCONNECT U33/Y U37/B0 (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[1\]/Q U36/A (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[2\]/QN U36/B (0.000:0.000:0.000))
(INTERCONNECT U36/Y U35/A (0.000:0.000:0.000))
(INTERCONNECT U34/Y U35/B (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[1\]/Q U34/A (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[2\]/QN U34/B (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[1\]/Q U33/A (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[0\]/QN U33/B (0.000:0.000:0.000))
(INTERCONNECT U36/Y U32/A (0.000:0.000:0.000))
(INTERCONNECT i_control\/state_reg\[0\]/QN U31/A (0.000:0.000:0.000))
(INTERCONNECT U36/Y U31/B (0.000:0.000:0.000))
(INTERCONNECT U16/Y U30/A0 (0.000:0.000:0.000))
(INTERCONNECT U32/Y U30/A1 (0.000:0.000:0.000))
(INTERCONNECT U9/Y U30/B0 (0.000:0.000:0.000))
(INTERCONNECT U29/Y U30/B1 (0.000:0.000:0.000))
(INTERCONNECT U36/Y U29/A (0.000:0.000:0.000))
(INTERCONNECT U2/Y U29/B (0.000:0.000:0.000))
(INTERCONNECT i_opcode_1/C U28/A0 (0.000:0.000:0.000))
(INTERCONNECT i_opcode_2/C U28/A1 (0.000:0.000:0.000))
(INTERCONNECT U2/Y U28/B0 (0.000:0.000:0.000))
Plain Text Tab Width: 8 Ln 1, Col 1 INS
a102 ASIC syn mapped control_pad.sdf (~/ASI...


tty2:177 - Xmanager 4 [0.0]
Applications Places System
control_pad.sdf (~/ASIC/syn/mapped) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
control_pad.sdf
)
)
)
(CELL
(CELLTYPE "OAI21XL")
(INSTANCE U38)
(DELAY
(ABSOLUTE
(IOPATH A0 Y (0.106:0.106:0.106) (0.050:0.050:0.050))
(IOPATH A1 Y (0.107:0.107:0.107) (0.049:0.049:0.049))
(IOPATH B0 Y (0.053:0.054:0.054) (0.045:0.046:0.046))
)
)
)
(CELL
(CELLTYPE "OAI21XL")
(INSTANCE U37)
(DELAY
(ABSOLUTE
(IOPATH A0 Y (0.106:0.106:0.106) (0.050:0.050:0.050))
(IOPATH A1 Y (0.114:0.114:0.114) (0.056:0.056:0.056))
(IOPATH B0 Y (0.053:0.054:0.054) (0.045:0.046:0.046))
)
)
)
(CELL
(CELLTYPE "NOR2X1")
(INSTANCE U36)
(DELAY
(ABSOLUTE
(IOPATH A Y (0.138:0.139:0.139) (0.067:0.067:0.067))
(IOPATH B Y (0.151:0.151:0.151) (0.079:0.080:0.080))
)
)
)
(CELL
(CELLTYPE "NOR2XL")
(INSTANCE U35)
(DELAY
(ABSOLUTE
(IOPATH A Y (0.065:0.068:0.068) (0.036:0.036:0.036))
(IOPATH B Y (0.067:0.067:0.067) (0.030:0.030:0.030))
Plain Text Tab Width: 8 Ln 1, Col 1 INS
a102 ASIC syn mapped control_pad.sdf (~/ASI...