

Lab2

Lab：开源EDA工具全流程复现

- Yosys: 实现Verilog代码到AIG的转换
- iMap和ABC可以实现AIG到netlist的转换
- 鹏城开发的后端工具iEDA，即netlist到版图GDS-II的转换

提供了一个简单的CPU设计，Verilog源码。（具体见附件的压缩包）
可用该CPU设计，去跑通以上工具流程（从Verilog到版图）

在上述过程中，做好word笔记，最终的汇报包括各步骤的具体功能描述、执行输出、遇到的问题 and 解决方式等。

实验报告

针对开源EDA工具的全流程执行。实验报告具体包括：

1. Verilog RTL用例的功能、结构简单分析。
2. 利用yosys等执行逻辑综合的过程，包括TCL命令、执行结果的截图、生成网表的报告（时延/资源使用量等）
3. 利用iEDA执行布局布线时序分析等中间过程和最终结果，包括TCL命令、命令的功能含义、执行结果（包括QoR报告结果，以及具体版图截图）。版图可以下载到本机查看并截图。
4. 必要的分析。

一、RTL用例分析_CPU

模块	功能
alu	算数逻辑单元，接受 8 位的数据和累加器输入，根据opcode执行不同的操作 PASS0、PASS1、PASS6、PASS7：直接将 accum 通过不同指令输出。 ADD：执行 accum + data 的加法。 AND：执行 accum & data 的按位与操作。 XOR：执行 accum ^ data 的按位异或操作。 PASSD：将 data 直接通过到输出。
clk	时钟生成模块。该模块生成一个50 MHz的方波时钟信号 clk
control	控制单元模块，用于管理多周期 CPU 的指令执行过程。该模块会基于指令操作码 (opcode) 和状态控制信号，在不同的时钟周期内生成适当的控制信号，以执行不同的指令操作，以完成从指令获取、解码、执行到存储的完整过程
counter	5 位的计数器模块 counter。该模块能够实现异步复位、同步加载预设值和计数功能

模块	功能
cpu	顶层 CPU 模块的设计，包含各个子模块的实例化和连接，形成了一个简单的多周期 CPU 控制单元 control 算数逻辑单元 ALU 累加器寄存器，指令寄存器 register 选择多路复用器 scale_mux 程序计数器 counter
diffrr	实现带异步复位的D触发器
mem	实现了一个32字节大小的内存单元，使用地址和读写信号来控制数据的存取。提供了一种简单的、通过地址访问的存储器接口，能够根据控制信号读取或写入数据，同时支持输出数据的高阻态以模拟双向数据总线
mux	2:1 多路复用器 (MUX)
register	8 位寄存器，它能够在时钟信号的上升沿根据加载信号的状态存储数据
scale_mux	可配置大小的 2:1 多路复用器 (MUX)，使用parameter size控制选择数据大小
full_chip	芯片的顶层设计模块，负责将 CPU 内部信号通过端口接口连接到外部

RTL设计实现了一个基础的 CPU 架构，能够处理简单的算术和逻辑运算，支持指令的读取、执行和存储。

二、Yosys逻辑综合

linux环境设置：

```
export LD_LIBRARY_PATH=$(echo $LD_LIBRARY_PATH | tr ':' '\n' | grep -v 'xilinx' | paste -sd ':')
```

排除Xilinx 工具环境影响

Yosys TCL命令

make syn

yosys.tcl 解析

1. 参数设置-设计名、文件路径、库文件、时钟频率
2. yosys 主运行过程
 1. 导入yosys
 2. 读取库文件
 3. 读取Verilog
 4. 对顶层设计进行合成 `synth -top $DESIGN`
 5. 设计优化
 6. 技术映射
3. 后处理

1. 将未定义值替换为0 `setundef -zero`
2. 分割网络，解决不需要的复合赋值语句 `splitnets`
3. 插入缓冲器单元
4. 清理未使用的单元和线
4. 报告和输出
 1. synth_check.txt 检查报告
 2. synth_stat.txt 统计报告
 3. 写入指定网表文件

执行结果

- full_chip.netlist.v
- synth_check.txt
- synth_stat.txt

```

=== design hierarchy ===
full_chip      1
  cpu          1
    $paramod\scale_mux\size=5  1
    alu         1
    control     1
    counter     1
    register    2
      dffr      8
      mux       8

Number of wires:      529
Number of wire bits:  648
Number of public wires: 271
Number of public wire bits: 390
Number of memories:   0
Number of memory bits: 0
Number of processes:  0
Number of cells:      374

```

- yosys.log

iSTA简单时序评估

- SDC文件
- sta.tcl

make sta

```
`LD_LIBRARY_PATH=bin/ ./bin/iSTA $(PROJ_PATH)/sta.tcl $(DESIGN) $(SDC_FILE)
$(NETLIST_V)
```

- 指定动态链接库: LD_LIBRARY_PATH=bin/
- 可执行工具: ./bin/iSTA
- TCL脚本 + 参数 (DESIGN、SDC_FILE、NETLIST_V)

```
set_design_workspace $RESULT_DIR
read_netlist $NETLIST_V
read_liberty $LIB_FILES
link_design $DESIGN
read_sdc $SDC_FILE
report_timing
```

sta输出

- full_chip_hold.skew：hold下时钟偏斜

Generate the report at 2024-11-01T03:11:51, GitVersion: d9c76846ce6e28304fe193503daa05d7efa8c4cc.
Clock: core_clock

Clock Pin	Latency	Skew	
u_cpu/ir/G1[7].U2/_1_:CK (DRNQUHDV1)	0.000		rp-+
u_cpu/ir/G1[7].U2/_1_:CK (DRNQUHDV1)	0.000	0.000	rp-+
u_cpu/ir/G1[4].U2/_1_:CK (DRNQUHDV1)	0.000		rp-+
u_cpu/ir/G1[4].U2/_1_:CK (DRNQUHDV1)	0.000	0.000	rp-+
u_cpu/ac/G1[7].U2/_1_:CK (DRNQUHDV1)	0.000		rp-+
u_cpu/ac/G1[7].U2/_1_:CK (DRNQUHDV1)	0.000	0.000	rp-+

- full_chip_setup.skew: setup下时钟偏斜

Generate the report at 2024-11-01T03:11:51, GitVersion: d9c76846ce6e28304fe193503daa05d7efa8c4cc.
Clock: core_clock

Clock Pin	Latency	Skew	
u_cpu/ctl1/_101_:CK (DRNQUHDV1)	0.000		rp-+
u_cpu/ir/G1[7].U2/_1_:CK (DRNQUHDV1)	0.000	0.000	rp-+
u_cpu/ctl1/_101_:CK (DRNQUHDV1)	0.000		rp-+
u_cpu/ir/G1[4].U2/_1_:CK (DRNQUHDV1)	0.000	0.000	rp-+
u_cpu/ac/G1[0].U2/_1_:CK (DRNQUHDV1)	0.000		rp-+
u_cpu/ac/G1[7].U2/_1_:CK (DRNQUHDV1)	0.000	0.000	rp-+

- full_chip.cap: 电容违规

Generate the report at 2024-11-01T03:11:51, GitVersion: d9c76846ce6e28304fe193503daa05d7efa8c4cc.

Net / InstPin	MaxCap	Cap	CapSlack	CellPort	Remark
clock					
u_cpu/ir/G1[7].U2/_1_:CK	NA/NA	0.002r/0.002f	NA/NA	DRNQUHDV1/CK	
clock					
u_cpu/ac/G1[5].U2/_1_:CK	NA/NA	0.002r/0.002f	NA/NA	DRNQUHDV1/CK	
clock					
u_cpu/ctl1/_090_:CK	NA/NA	0.002r/0.002f	NA/NA	DSNQUHDV1/CK	

- full_chip.fanout: 扇出违规

```
Generate the report at 2024-11-01T03:11:51, GitVersion: d9c76846ce6e28304fe193503daa05d7efa8c4cc.
```

Net / InstPin	MaxFanout	FanLoad	FanLoadSlack	CellPort	Remark
addr_pin[4]					
u_cpu/smx/_36_:Z	NA	0	NA	BUFUHDV1/Z	
u_cpu/alu1/_027_					
u_cpu/alu1/_187_:ZN	NA	1	NA	AOI221UHDV0P4/ZN	
u_cpu/ac/G1[5].U1/_2_					
u_cpu/ac/G1[5].U1/_4_:Z	NA	1	NA	MUX2UHDV0P4/Z	

- full_chip.rpt: 时序分析报告

```
Generate the report at 2024-11-01T03:11:51, GitVersion: d9c76846ce6e28304fe193503daa05d7efa8c4cc.
```

Endpoint	Clock Group	Delay Type	Path Delay	Path Required	CPPR	Slack	Freq(MHz)
u_cpu/ac/G1[6].U2/_1_:D	core_clock	max	2.853f	1.823	0.000	-1.030	330.064
u_cpu/ac/G1[7].U2/_1_:D	core_clock	max	2.822f	1.823	0.000	-0.999	333.470
u_cpu/ac/G1[7].U2/_1_:D	core_clock	max	2.788r	1.791	0.000	-0.997	333.688
u_cpu/ctl1/_091_:D	core_clock	min	0.400f	0.008	0.000	0.392	NA
u_cpu/ctl1/_096_:D	core_clock	min	0.400f	0.008	0.000	0.392	NA
u_cpu/ctl1/_095_:D	core_clock	min	0.400f	0.008	0.000	0.392	NA

Clock	Delay Type	TNS
core_clock	max	-14.085
core_clock	min	0.000

- full_chip.trans: 转换时间违规

```
Generate the report at 2024-11-01T03:11:51, GitVersion: d9c76846ce6e28304fe193503daa05d7efa8c4cc.
```

Net / InstPin	MaxTranTime	TranTime	TranSlack	CellPort	Remark
u_cpu/alu1/_034_					
u_cpu/alu1/_111_:ZN	3.030r/3.030f	0.795r/0.426f	2.235r/2.604f	XNOR2UHDV0P4/ZN	
u_cpu/alu1/_034_					
u_cpu/alu1/_112_:I	3.030r/3.030f	0.795r/0.426f	2.235r/2.604f	INUHDV0P4/I	
u_cpu/alu1/_034_					
u_cpu/alu1/_126_:B1	3.030r/3.030f	0.795r/0.426f	2.235r/2.604f	AOI32UHDV0P4/B1	

目录

```

v bin
  i iSTA
  i libyaml-cpp.so.0.6
> cpu
> example
> result
> smic180
. .gitignore
glog_dump.log
M Makefile
i README.md
sta.tcl
yosys.tcl

```

执行结果

```
● [/home/huzibo/large/Work_1/Eda_Lab/lab2/yosys-sta/result/full_chip-500MHz]$ll
total 364
-rw-rw-r-- 1 huzibo huzibo 988 Nov 29 10:27 full_chip.cap
-rw-rw-r-- 1 huzibo huzibo 1028 Nov 29 10:27 full_chip.fanout
-rw-rw-r-- 1 huzibo huzibo 6554 Nov 29 10:27 full_chip_hold.skew
-rw-rw-r-- 1 huzibo huzibo 28653 Nov 29 10:27 full_chip.netlist.v
-rw-rw-r-- 1 huzibo huzibo 37940 Nov 29 10:27 full_chip.rpt
-rw-rw-r-- 1 huzibo huzibo 6554 Nov 29 10:27 full_chip_setup.skew
-rw-rw-r-- 1 huzibo huzibo 1088 Nov 29 10:27 full_chip.trans
-rw-rw-r-- 1 huzibo huzibo 44253 Nov 29 10:27 full_chip.v
-rw-rw-r-- 1 huzibo huzibo 2591 Nov 29 10:27 synth_check.txt
-rw-rw-r-- 1 huzibo huzibo 10559 Nov 29 10:27 synth_stat.txt
-rw-rw-r-- 1 huzibo huzibo 212453 Nov 29 10:27 yosys.log
```

iEDA布线布局

1. 将本次 cpu 的 netlist 和 sdc 文件拷贝到相应位置
2. 修改 db_path_setting.tcl 文件
3. 其他修改

run_iEDA.py

Py文件执行了多个tcl脚本

1. run_iFP.tcl: 进行布局规划 (Floorplan) , 确定芯片的基本布局和区域分配。
 2. run_iNO_fix_fanout.tcl: 修复扇出 (Fanout) 问题, 确保信号在电路中的传输不会受到影响。
 3. run_iPL.tcl: 进行放置 (Placement) , 将电路元件放置在芯片上以优化性能。
 4. run_iCTS.tcl: 进行时钟树合成 (Clock Tree Synthesis) , 优化时钟信号的分配以减少延迟。
 5. run_iTO_drv.tcl: 修复驱动 (Driver) 问题, 确保电路的驱动能力满足设计要求。
 6. run_iTO_hold.tcl: 优化保持时间 (Hold Time) , 确保信号在时钟边缘到达时的稳定性。
 7. run_iPL_legalization.tcl: 进行放置合法化 (Placement Legalization) , 确保放置的元件符合设计规则。
 5. run_iRT.tcl: 进行路由 (Routing) , 连接电路元件以形成完整的电路路径。
 6. run_iPL_filler.tcl: 添加填充物 (Filler) , 确保芯片的布局密度符合制造要求。
 10. run_def_to_gds_text.tcl: 将设计交换格式 (DEF) 转换为GDSII格式, 准备进行制造。
- 这些脚本在芯片设计流程中起着关键作用, 确保设计的有效性和可制造性。

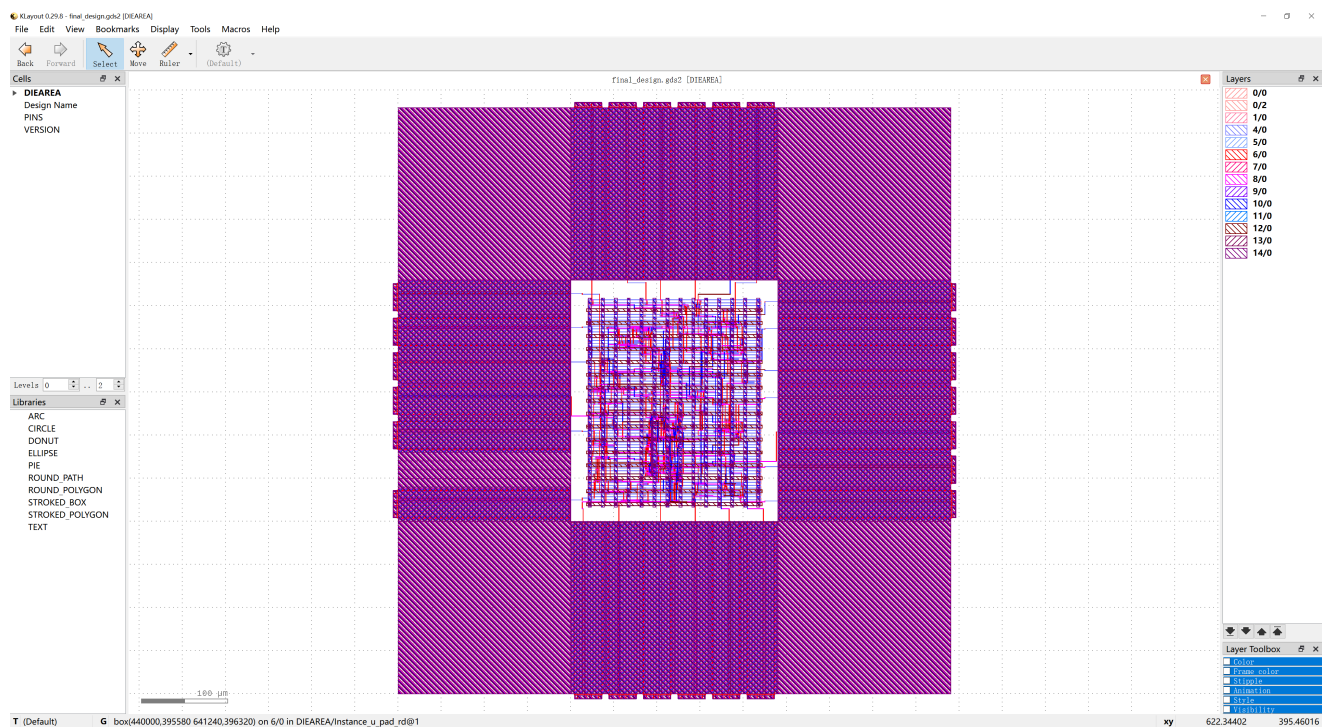
版图

Create file success (./result/final_design.gds2)

Write COMPONENTS success. 708 / 708

No FILLS ...

Write NETS success. 361 / 361



附件

makefile

文件依赖关系

Makefile 中，文件的依赖关系通过目标和依赖项的定义来建立

【make syn】

- **syn: \$(NETLIST_V)**
 - syn 依赖于 \$(NETLIST_V)
- **\$(NETLIST_V): \$(RTL_FILES) yosys.tcl**
 - \$(NETLIST_V) 依赖于 RTL 文件和 Yosys 脚本

【make sta】

- **sta: \$(TIMING_RPT)**
 - sta 依赖于 \$(TIMING_RPT)
- **\$(TIMING_RPT): \$(SDC_FILE) \$(NETLIST_V)**
 - \$(TIMING_RPT) 依赖于 \$(SDC_FILE) \$(NETLIST_V)