
目录

实验三 布图规划 1	2
实验目的	2
实验内容	2
版图基础知识	2
IP 核基础知识	4
IO 引脚基础知识	5
实验步骤	6
IP 核（LC3 Memory 模块）的置换	6
版图大小设置实验	9
IO 引脚设置实验	13
实验练习	15

实验三 布图规划 1

实验目的

- 1) 理解版图基础知识，掌握布图规划中的版图设置操作
- 2) 理解 IP 核基础知识，掌握 IP 核置换操作
- 3) 理解 IO 引脚基础知识，掌握 IO 引脚摆放操作实验内容

实验内容

布图规划将逻辑描述（综合结果）进行第一步物理化，其目标主要有以下几点：

- 1) 确定芯片版图大小
- 2) 决定 IO（Input&Output）引脚在版图上的位置
- 3) 在版图上摆放物理单元 Tapcell
- 4) 将 IP 核放置在芯片版图上
- 5) 布置电源网络

版图基础知识

芯片版图是真实集成电路物理情况的平面几何形状描述。芯片版图是集成电路设计中最底层步骤物理设计的成果，物理设计通过布局、布线技术将逻辑综合的成果——门级的网表转换成物理版图文件，这个文件包含了各个硬件单元在芯片上的形状、面积和位置信息。这个芯片版图十分复杂，不过我们只需先关注版图的尺寸大小。

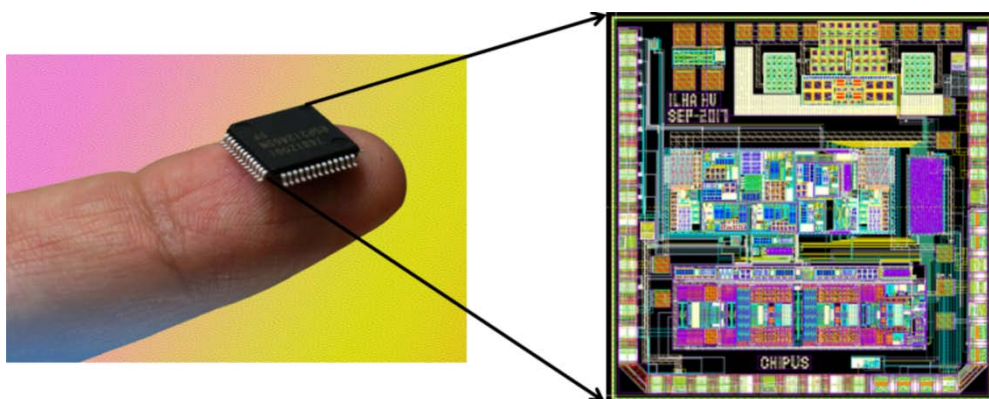


图 1 芯片的版图描述

版图的大小确定了最终芯片最终的尺寸。本实验面临的第一个问题是如何设置版图的大小？设置的依据是什么？

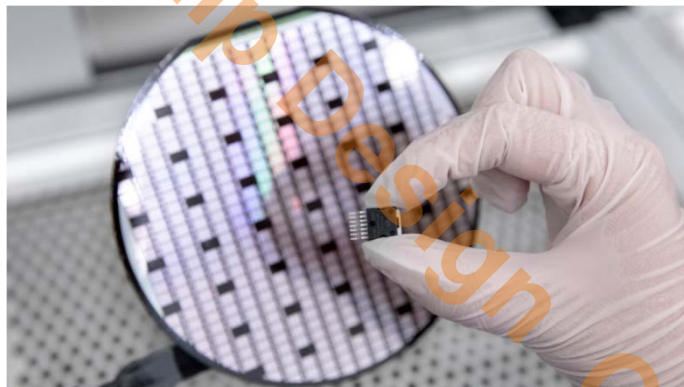


图 2 晶圆与芯片

晶圆是半导体晶体圆形片的简称，是生产芯片所用的载体。以华为麒麟 990 5G 芯片举例，12 英寸晶圆如果 100%利用的话能生产 700 块麒麟芯片。出于成本的考虑，芯片的面积越小，每张晶圆上产出的芯片数量将增大，从而平均到每个芯片上的成本会降低。然而，如果设定版图的面积太小，则会造成后续布线时拥塞程度高，从而导致长周期的设计迭代。一个合理的版图大小设定是在保证布线的同时尽量节约产品成本。

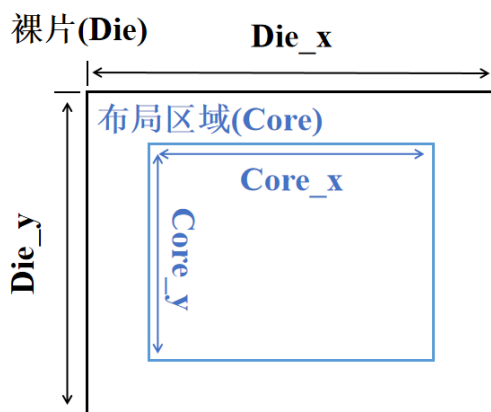


图 3 版图大小描述

裸片（Die）的大小接近实际生活中接触的芯片大小，布局区域（Core）规定了物理后端设计摆放单元器件的范围，实验描述的版图设置是设置 Die 的大小。总结版图大小的设置可以由以下两个方面进行考虑。

1) 利用率。当完成芯片前端代码编写以及综合步骤后，网表映射的单元总体的面积是已知的（将所有单元的面积进行加和），一般单元的总面积会占芯片面积的 50%~70%。

$$2) \text{ 芯片版图面积} = \frac{\text{单元面积总和}}{\text{利用率}}$$

设定版图大小调用布局布线工具进行评估。芯片版图的大小直接影响后端物理设计的后续布局布线流程，如版图过小，单元之间的间隙不足单元的引脚可能过于密集导致绕线困难。不同的布局布线工具对小版图的优化力度不一样，例如目前国外 EDA 三巨头 Synopsys、Cadence、Mentor Graphics 商业公司旗下的布局布线产品能够在有限的版图空间中跑通布局布线，而开源的布局布线工具则不一定能够完成任务，核心竞争力在于算法。因此，判断芯片版图大小是否合理，可以先调用布局布线工具查看在当前大小版图下是否能够完成芯片物理后端操作。

除了面积最小化能节省芯片制造成本外，另外一个目标是尽可能地使版图边框的长宽比约等于 1:1。由于受到制造和封装尺寸的限制，一个正方形芯片（长宽比 ≈ 1 ）会比一个非正方形芯片更好。

IP 核基础知识

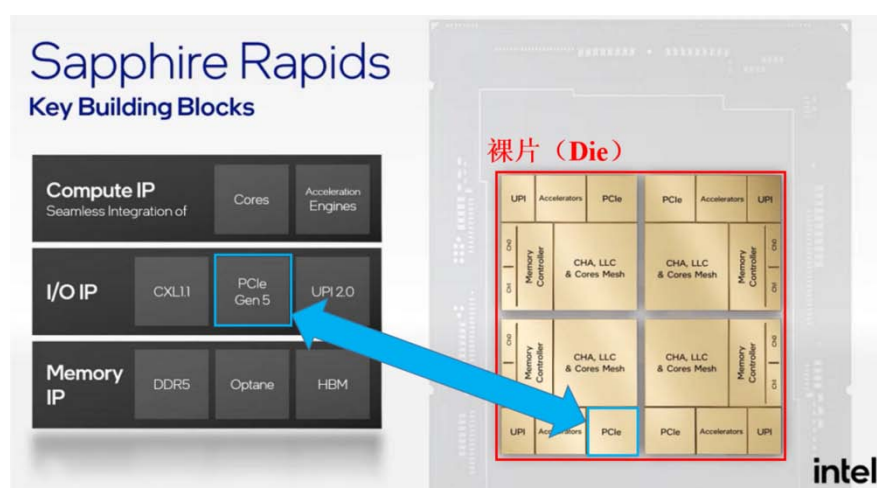


图 4 芯片设计中的 IP 核

IP 核，全称知识产权核，是在集成电路的可重用设计方法学中，指某一方提供的、形式为逻辑单元，芯片设计的可重用模块。IP 核通常已经通过了设计验证，设计人员以 IP 核为基础进行设计，可以缩短设计所需的周期。IP 核可以通过协议由一方提供给另一方，或由一方独自占有。IP 核的概念源于产品设计的专利证书和源代码的著作权等。设计人员能够以 IP 核为基础进行设计，以减少设计周期。

IP 核分为软核、硬核和固核。软核通常是与工艺无关、具有寄存器传输级硬件描述语言描述的设计代码（编写前端使用的 Chisel 语言最终转化成的寄存器传输级语言），可以进行后续设计。硬核是前者通过后端物理设计中的逻辑综合、

布局、布线之后的一系列表征文件，具有特定的工艺形式、物理实现方式（如实验过程中查看布局布线状态的 GDSII 文件）。固核则通常介于上面两者之间，它已经通过功能验证、时序分析等过程，设计人员可以从逻辑门级网表的形式获取。

裸片(Die)

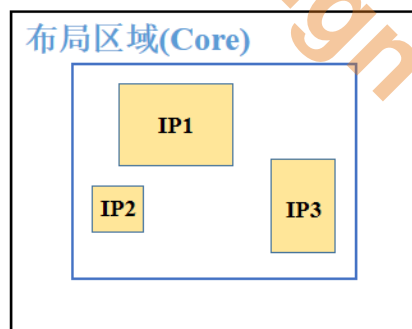


图 5 IP 核在芯片版图中

IP 核在实际后端物理设计过程中的表现形式是“占面积很大的单元”，因此也称为宏单元。《实验四 布图规划 2》中将会详细描述宏单元的位置摆放。

IO 引脚基础知识

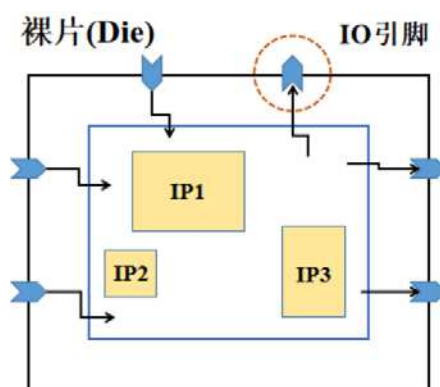


图 6 IO 引脚在芯片版图中

引脚是指从集成电路（芯片）内部电路引出与外围电路的接线，引脚构成了芯片与外界进行交互的接口，引脚与芯片内部的单元同样存在连线。引脚的摆放。引脚摆放的位置确定需要综合考虑与内部单元的连线，保证信号从芯片内部传递到外部时其路径最短，同时要避免连线交叉，方便封装基板的制作，减少基板上的走线层数，从而降低封装的成本。因此摆放引脚的依据是：

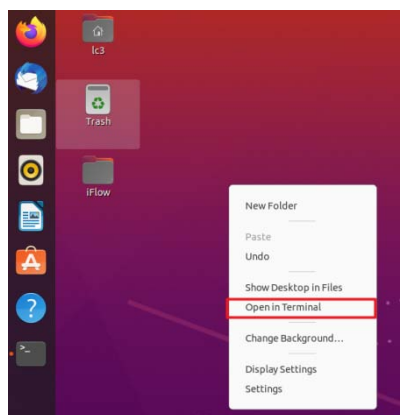
- 1) IO 引脚与芯片内部单元的连线线长短，换言之 IO 引脚的摆放需要靠近所连接的单元
- 2) 不同 IO 引脚与芯片内部单元的连线尽量避免线的交叉

当前实际摆放 IO 引脚的过程中，更依赖于工程师对芯片设计的经验。电子设计自动化（EDA）工具在布图规划中由于未进行布局，内部单元的位置还未确定，因此 IO 引脚与内部单元的线长也不能确定。此时可以通过调用布局工具进行单元位置的预摆放，从而给 IO 引脚的摆放提供评价指标。

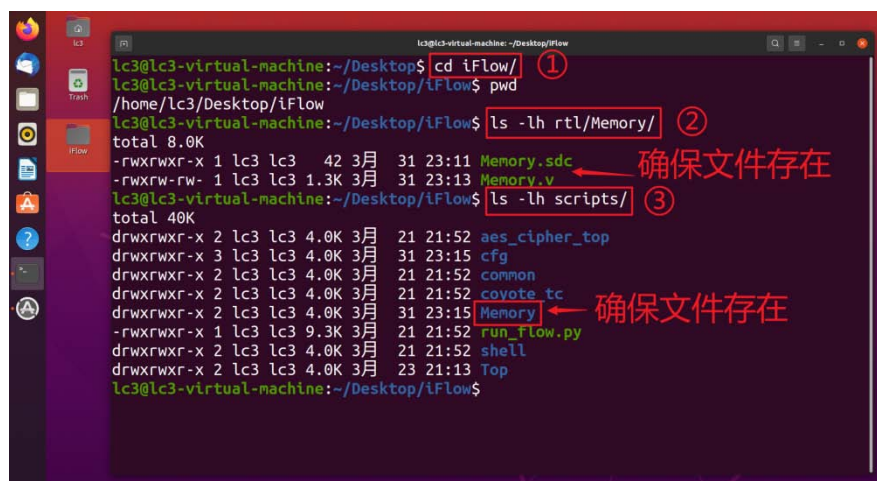
实验步骤

IP 核（LC3 Memory 模块）的置换

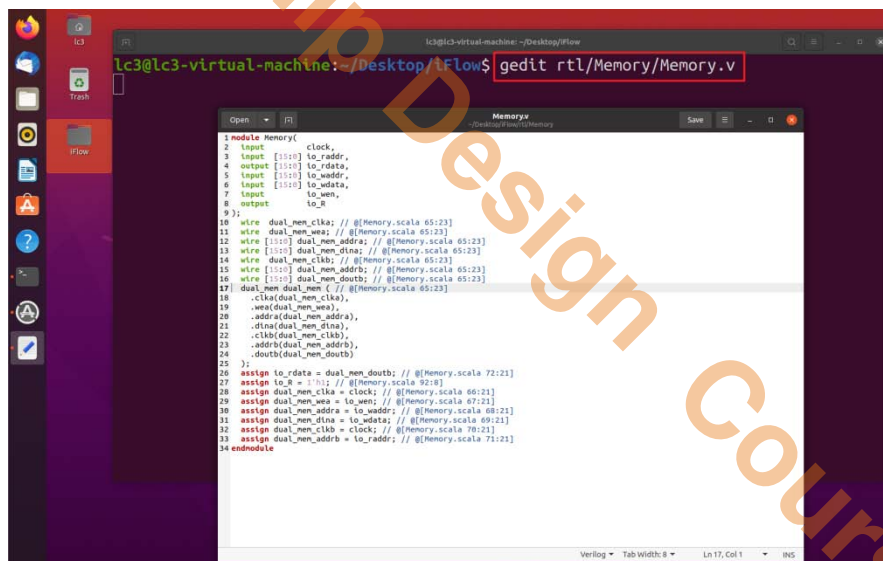
- 1) 虚拟机开机，并打开终端



- 2) 检查 Memory 是否已配置。若下图中的文件不存在，需返回环境配置实验



- 3) LC3 模块中，需要对 Memory 模块进行 IP 核置换。使用 gedit 工具打开 Memory.v 文件



4) 在打开的 Memory.v 文件中进行如下替换操作。首先通过快捷键“ctrl”+“f”打开搜索框，查找对应 IP 的名字“dual_mem”



5) 将设计中的 IP 名称及端口更换为工艺库内的 IP 名和端口。将红色矩形区域内的代码开头用“//”注释，并在下方填入 sky130_sram_1rw1r_44x64_8 内容

```

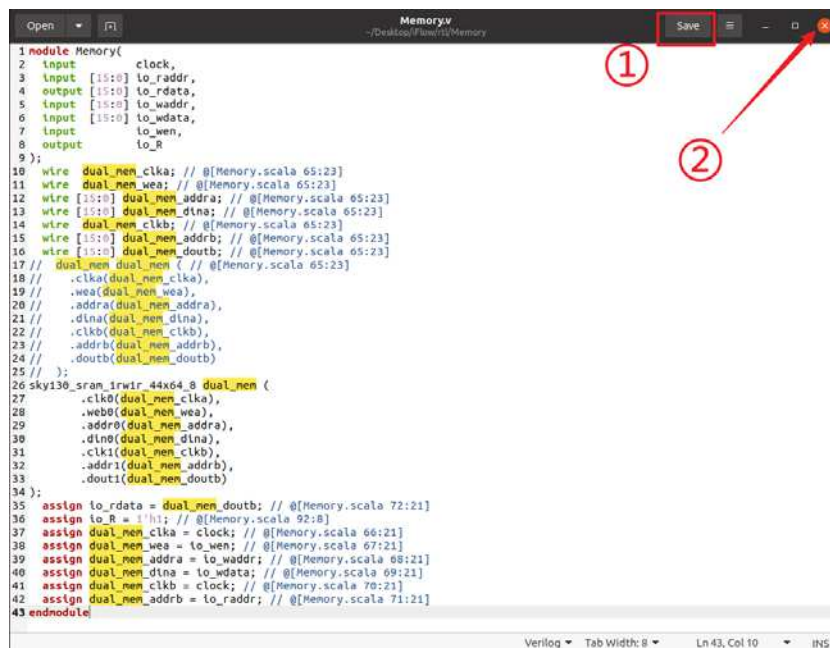
15 wire [15:0] dual_mem_addrb; // @[Memory.scala 65:
16 wire [15:0] dual_mem_doutb; // @[Memory.scala 65:
17 // dual_mem dual_mem ( // @[Memory.scala 65:23]
18 // .clka(dual_mem_clka),
19 // .wea(dual_mem_wea),
20 // .addra(dual_mem_addra),
21 // .dina(dual_mem_dina),
22 // .clkb(dual_mem_clkb),
23 // .addrb(dual_mem_addrb),
24 // .doutb(dual_mem_doutb)
25 // );
26 sky130_sram_1rw1r_44x64_8 dual_mem (
27   .clk0(dual_mem_clka),
28   .web0(dual_mem_wea),
29   .addr0(dual_mem_addra),
30   .din0(dual_mem_dina),
31   .clk1(dual_mem_clkb),
32   .addr1(dual_mem_addrb),
33   .dout1(dual_mem_doutb)
34 );

```

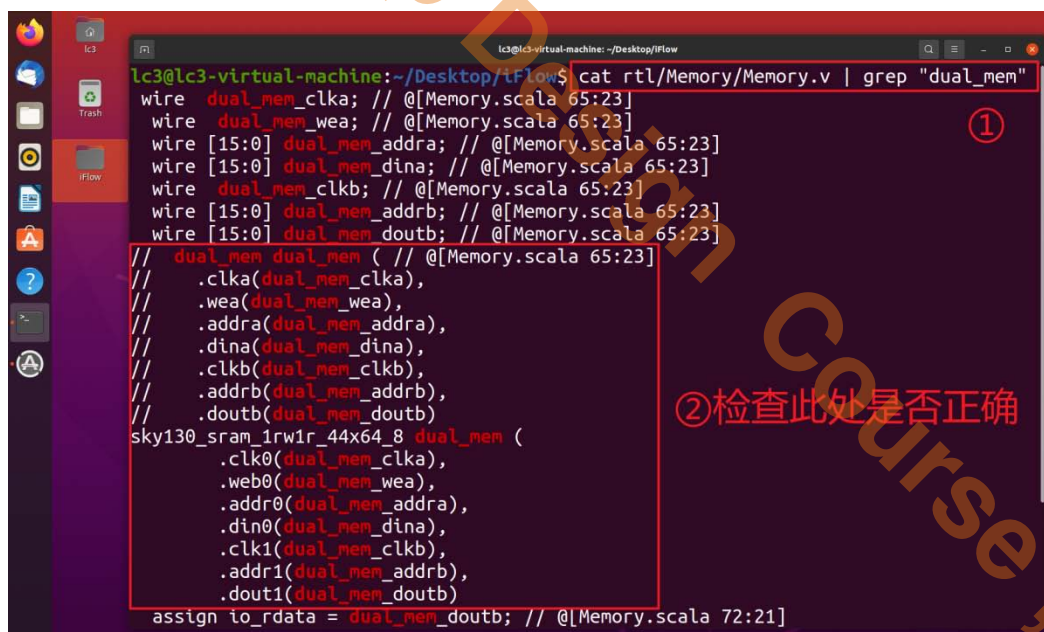
注：实验采用的工艺库 Sky130 中的有 4 种 IP，这些 IP 的接口都统一为下
sky130_sram_1rw1r_*

- 时钟端口 clk0 和 clk1
- 地址端口 addr0 和 addr1
- 写使能端口 web0
- 数据输入端口 din0
- 数据输出端口 dout0 和 dout1
- 片选端口 csb0 和 csb1

6) 更换完 IP 后保存并退出



7) 通过 Cat 命令查看 Memory.v 内容是否修改成功



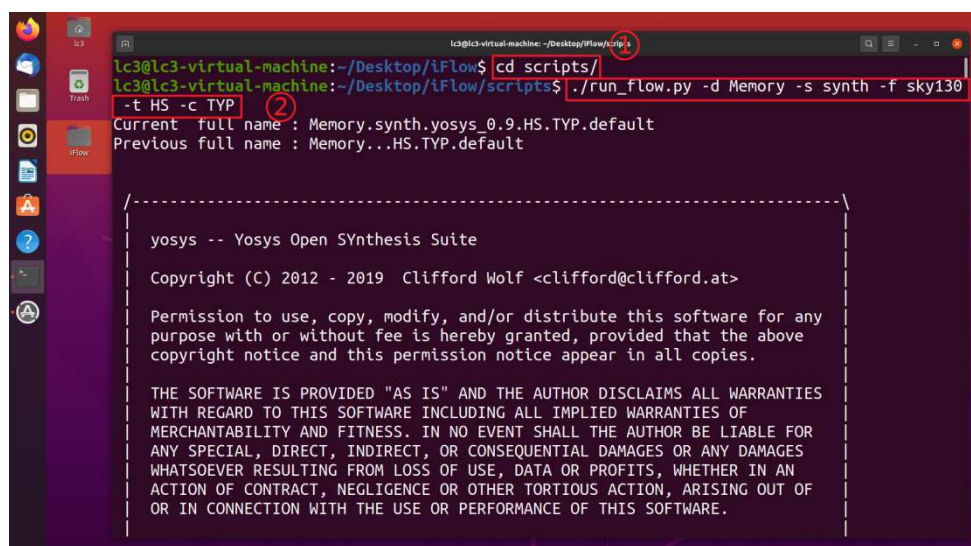
```
lc3@lc3-virtual-machine: ~/Desktop/iFlow$ cat rtl/Memory/Memory.v | grep "dual_mem"
wire dual_mem_clka; // @[Memory.scala 65:23]
wire dual_mem_wea; // @[Memory.scala 65:23]
wire [15:0] dual_mem_addra; // @[Memory.scala 65:23]
wire [15:0] dual_mem_dina; // @[Memory.scala 65:23]
wire dual_mem_clkb; // @[Memory.scala 65:23]
wire [15:0] dual_mem_addrb; // @[Memory.scala 65:23]
wire [15:0] dual_mem_doutb; // @[Memory.scala 65:23]
// dual_mem dual_mem ( // @[Memory.scala 65:23]
//   .clka(dual_mem_clka),
//   .wea(dual_mem_wea),
//   .addra(dual_mem_addra),
//   .dina(dual_mem_dina),
//   .clkb(dual_mem_clkb),
//   .addrb(dual_mem_addrb),
//   .doutb(dual_mem_doutb)
sky130_sram_1rw1r_44x64_8 dual_mem (
  .clk0(dual_mem_clka),
  .web0(dual_mem_wea),
  .addr0(dual_mem_addra),
  .din0(dual_mem_dina),
  .clk1(dual_mem_clkb),
  .addr1(dual_mem_addrb),
  .dout1(dual_mem_doutb)
assign io_rdata = dual_mem_doutb; // @[Memory.scala 72:21]
```

①

②检查此处是否正确

版图大小设置实验

1) 首先完成布图规划前的所有步骤：综合



```
lc3@lc3-virtual-machine: ~/Desktop/iFlow$ cd scripts/
lc3@lc3-virtual-machine: ~/Desktop/iFlow/scripts$ ./run_flow.py -d Memory -s synth -f sky130
-t HS -c TYP
Current full name : Memory.synth.yosys_0.9.HS.TYP.default
Previous full name : Memory...HS.TYP.default

-----
yosys -- Yosys Open Synthesis Suite

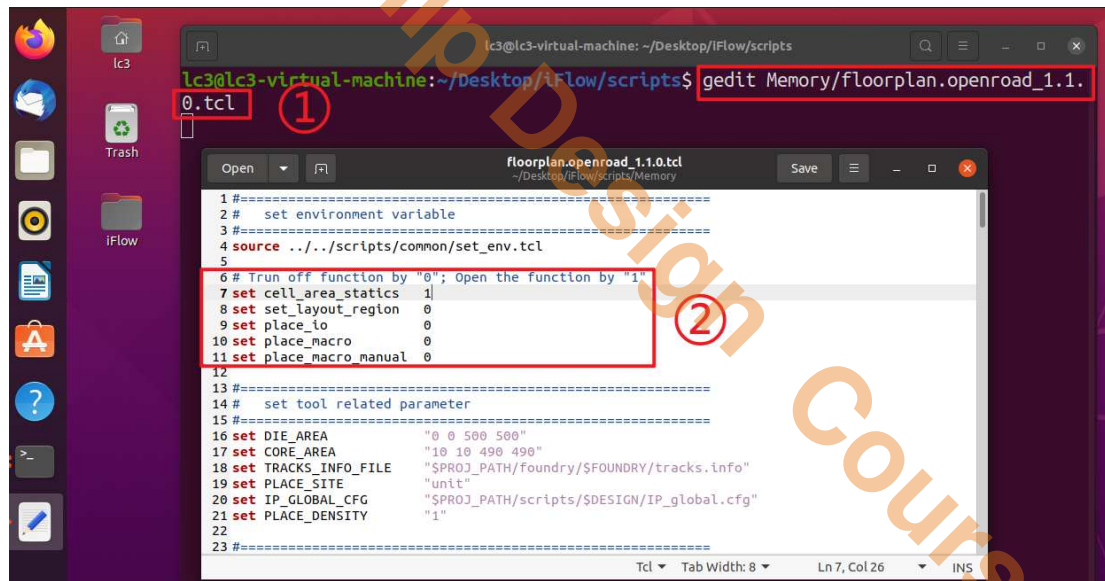
Copyright (C) 2012 - 2019 Clifford Wolf <clifford@clifford.at>

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
-----
```

2) 计算芯片版图大小

进行单元总面积统计



保存后退出。

对②中的设置进行解释

```
7 set cell_area_statics 1
8 set set_layout_region 0
9 set place_io 0
10 set place_macro 0
11 set place_macro_manual 0
```

“cell_area_statics” 控制是否进行单元面积统计，若为 1 则打开，若为 0 则跳过该步骤。

“set_layout_region” 控制是否进行版图区域大小设置，若为 1 则打开，若为 0 则跳过该步骤。

“place_io” 控制是否摆放芯片边缘的 io 引脚，若为 1 则打开，若为 0 则跳过该步骤。

“place_macro” 控制是否自动摆放宏单元，若为 1 则打开，若为 0 则跳过该步骤。

“place_macro_munual” 控制是否手动摆放宏单元，若为 1 则打开，若为 0 则跳过该步骤。

该设置十分重要，后续步骤会频繁出现。

```
lc3@lc3-virtual-machine: ~/Desktop/iFlow/scripts
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ ./run_flow.py -d Memory -s floorplan
-f sky130 -t HS -c TYP
Current full name : Memory.floorplan.openroad_1.1.0.HS.TYP.default
Previous full name : Memory.synth.yosys_0.9.HS.TYP.default

OpenROAD 1.1.0 9295a533ff
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>

This is free software, and you are free to change and redistribute it
under certain conditions; type 'show_copyright' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show_warranty'.

v, line 19 instance dual_mem port addr1 size 6 does not match net size 16.
Warning: /home/lc3/Desktop/iFlow/result/Memory.synth.yosys_0.9.HS.TYP.default/Memory.
v, line 19 instance dual_mem port din0 size 44 does not match net size 16.
Warning: /home/lc3/Desktop/iFlow/result/Memory.synth.yosys_0.9.HS.TYP.default/Memory.
v, line 19 instance dual_mem port dout1 size 44 does not match net size 16.

=====
report total insts area
=====
105717.8434 (um^2)
=====

lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$
```

芯片利用率取 50%，芯片版图面积 = $105717.8434 / 0.5 = 211435.6868(\text{um}^2)$

根据芯片长宽比最好为 1:1，对芯片版图面积开平方，可得芯片尺寸：宽 459.821，高 459.821。实验示例暂设为 600×600 （支持芯片版图尺寸设置宽 460，高 460 以上保证程序不出错）。

3) 修改版图配置文件，同时修改版图及布局区域大小

```
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ gedit Memory/floorplan.openroad_1.1.0.tcl

floorplan.openroad_1.1.0.tcl
~/Desktop/iFlow/scripts/Memory

1 #=====
2 # set environment variable
3 #=====
4 source ../../scripts/common/set_env.tcl
5
6 # Trun off function by "0". Open the function by "1"
7 set cell_area_statics 0
8 set set_layout_region 1
9 set place_io 0
10 set place_macro 0
11 set place_macro_manual 0
12
13 #=====
14 # set tool related parameter
15 #=====
16 set DIE_AREA "0 0 600 600"
17 set CORE_AREA "10 10 590 590"
18 set TRACKS_INFO_FILE "$PROJ_PATH/foandry/$FOUNDRY/tracks.info"
19 set PLACE_SITE "unit"
20 set IP_GLOBAL_CFG "$PROJ_PATH/scripts/$DESIGN/IP_global.cfg"
21 set PLACE_DENSITY "1"
22
23 #=====

Tcl Tab Width: 8 Ln 17, Col 41 INS
```

十分需要注意的是②中的“cell_area_statics”开关必须关掉，否则程序无法正常设置版图大小。

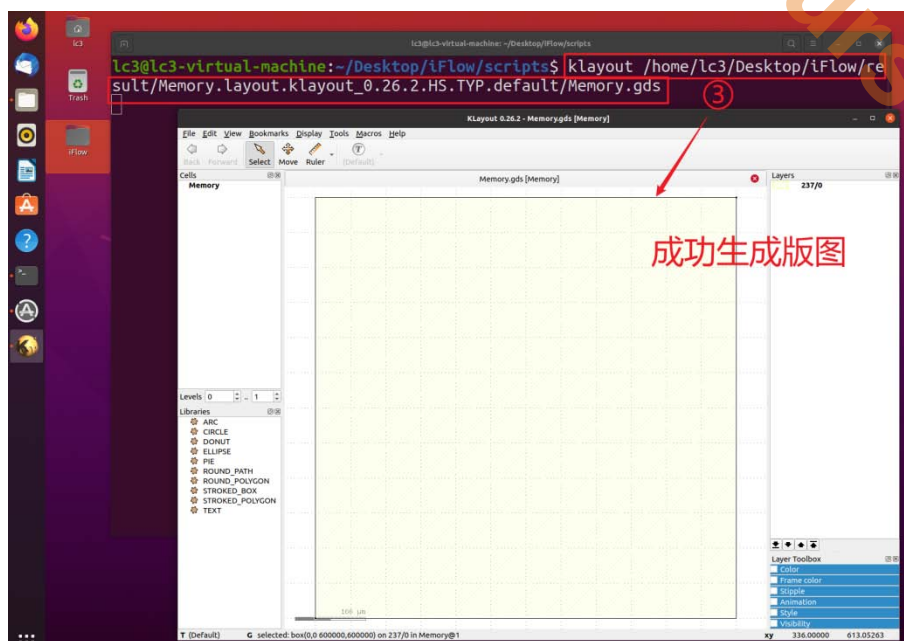
4) 查看结果

```
lc3@lc3-virtual-machine: ~/Desktop/iFlow/scripts
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ ./run_flow.py -d Memory -s floorpl
an,layout -f sky130 -t HS -c TYP
Current full name : Memory.floorplan.openroad_1.1.0.HS.TYP.default
Previous full name : Memory.synth.yosys_0.9.HS.TYP.default

OpenROAD 1.1.0 9295a533ff
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>
```

自动弹出的版图文件会有些缺失，推荐使用 **klayout** 打开该 **gds** 文件，屏幕输出的最后会指明版图文件的路径。

```
[INFO] All LEF cells have matching GDS cells
[INFO] Writing out GDS '/home/lc3/Desktop/iFlow/result/Memory.layout.klayout_0.26.2
HS.TYP.default/Memory.gds'
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$
```



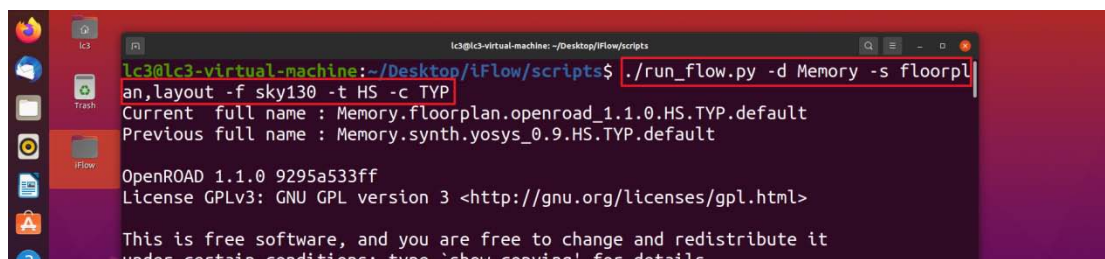


IO 引脚设置实验

1) 设置 IO 引脚脚本开关

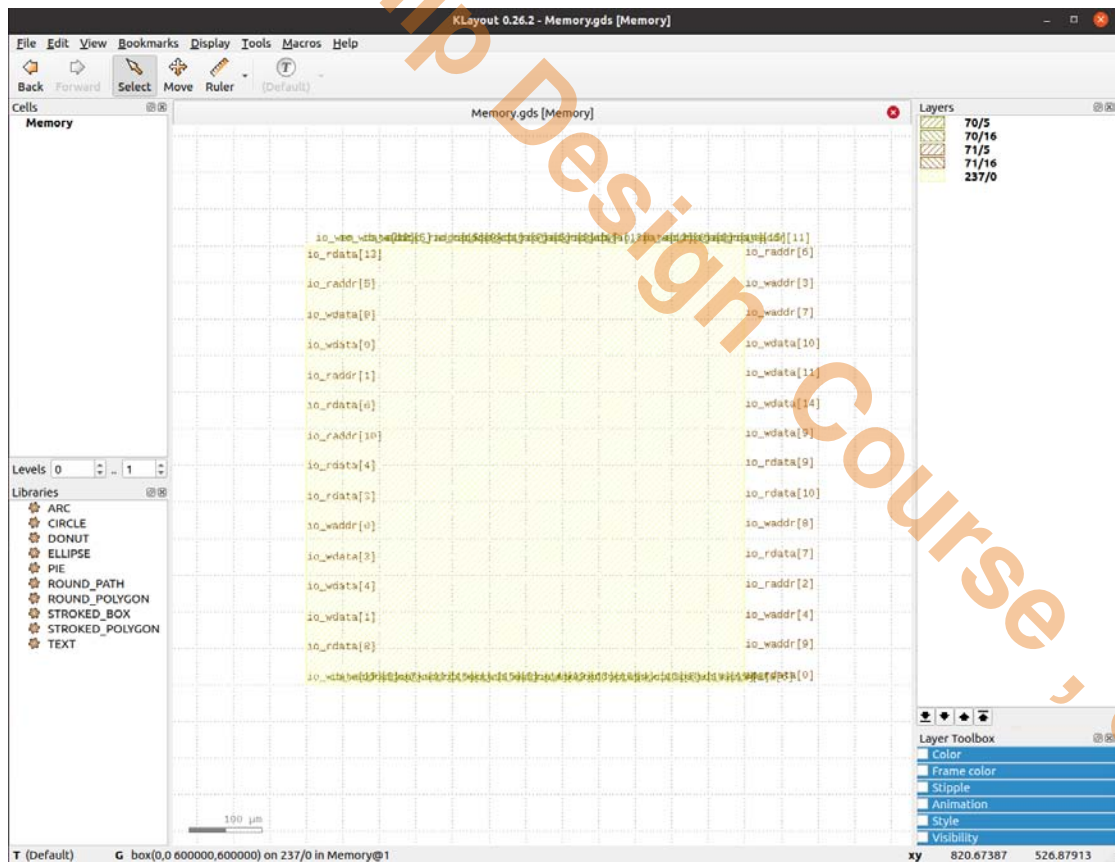


2) 运行 IO 引脚摆放脚本



3) 查看 GDSII 版图

若无法得到下图效果，可手动按照屏幕输出最后的文件路径打开 gds 文件
(具体可参照版图大小设置实验)



4) 统计线长指标评估 IO 引脚的摆放好坏

摆放 IO 后的线长统计信息评估

```

lc3@lc3-virtual-machine: ~/Desktop/iFlow/scripts
v, line 19 instance dual_mem port dout1 size 44 does not match net size 16.
Info: Added 174 rows of 1209 sites.

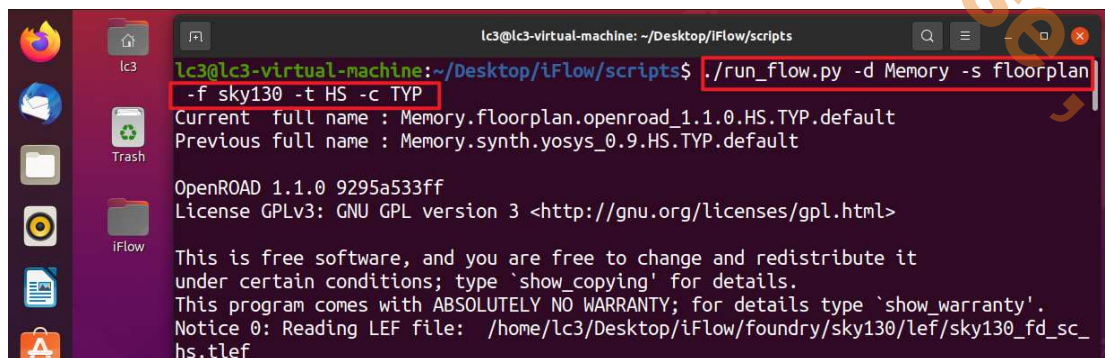
=====
Begin Place IO pins
-----
WARNING: force pin spread option has no effect when using random pin placement
> Running IO placement
* Num of slots          1438
* Num of I/O            67
* Num of I/O w/sink      3
* Num of I/O w/o sink   64
* Slots Per Section     200
* Slots Increase Factor  0.01
* Usage Per Section      0.8
* Usage Increase Factor  0.01
* Force Pin Spread       1

WARNING: running random pin placement
RandomMode Even
> IO placement done.

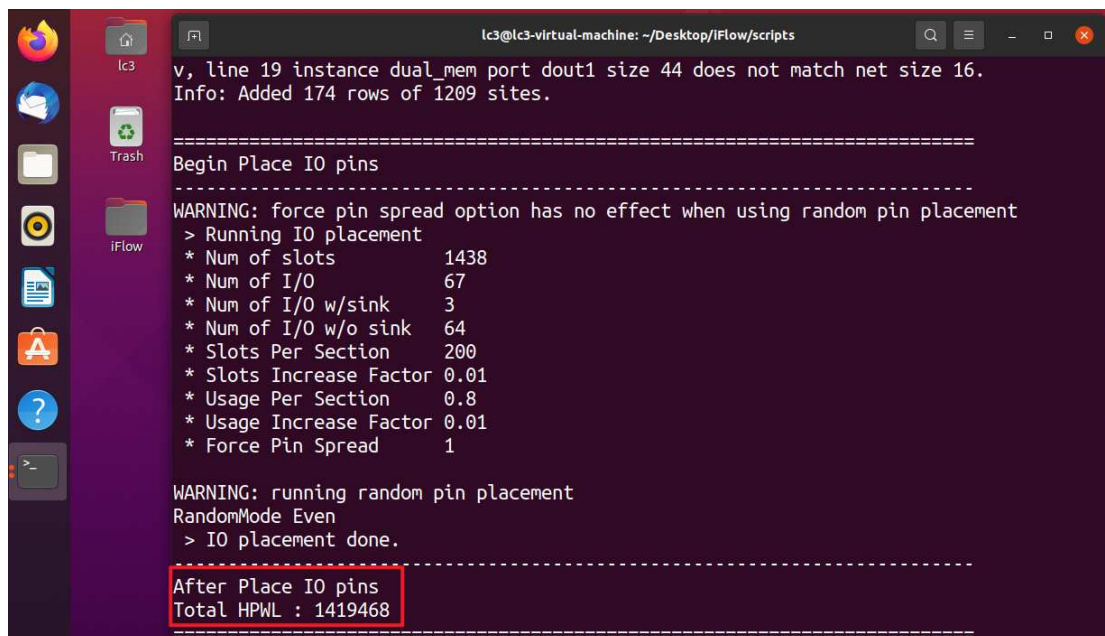
-----
After Place IO pins
Total HPWL : 2366972
=====

```

改动 IO 自动摆放的随机参数



得到的线长结果是：



实验练习

- 1) 自主改变版图大小，并用标尺在 GDSII 版图上进行测量验证

-
- 2) 在 IO 引脚摆放后输出的 GDSII 版图中，标出芯片的输入引脚
 - 3) 自主摆放 IO 引脚并进行结果评估