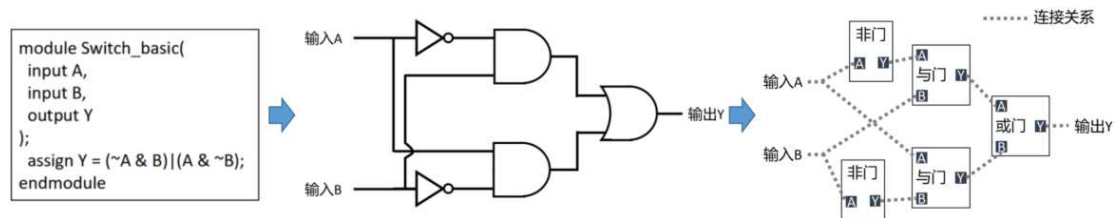

目录

实验二 综合	2
实验目的.....	2
实验内容.....	2
逻辑优化	2
工艺库介绍	4
单元置换	6
实验步骤.....	7
逻辑优化	7
查看工艺库	12
不同工艺库下的单元置换	15
查看网表	17
小结	17
作业	18

实验二 综合

综合的主要过程如下图所示，先将硬件描述语言转换为逻辑门电路，再将逻辑门电路转换为标准单元之间具有连接关系的电路图（网表，netlist）。



实验目的

- (1) 了解电路优化基本原理
- (2) 了解工艺库基本概念
- (3) 了解电路置换基本原理
- (4) 掌握使用综合工具查看电路图

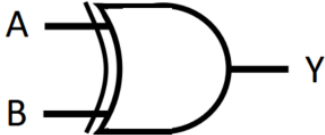
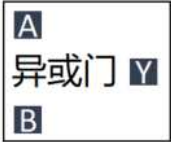
实验内容

逻辑优化

在实验一介绍了标准单元是实现逻辑门相同功能的器件。如下表所示，每一个标准单元对应于一个逻辑门。

逻辑门	标准单元

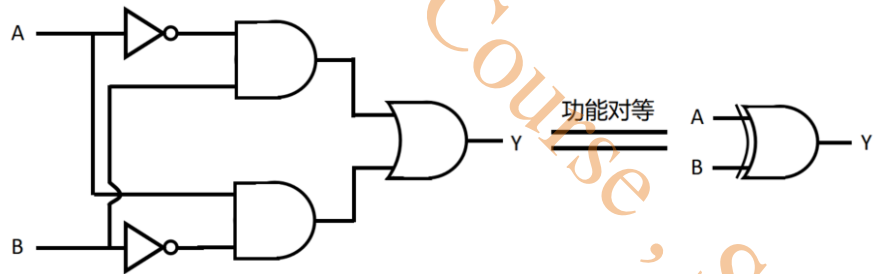
这里介绍了基本的三个逻辑门（非门、与门和或门）。除了这三个基础逻辑门外，这里再介绍一个异或逻辑门。

逻辑门	标准单元
	

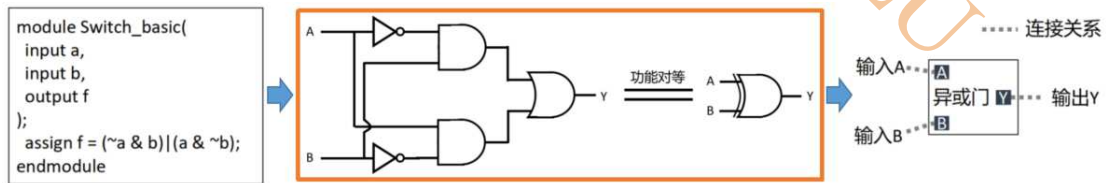
异或逻辑门的符号如上表所示，看起来像是在或门的左边加了一个曲线。它的真值表如下。

输入 A	输入 B	输出 Y
0	0	0
0	1	1
1	0	1
1	1	0

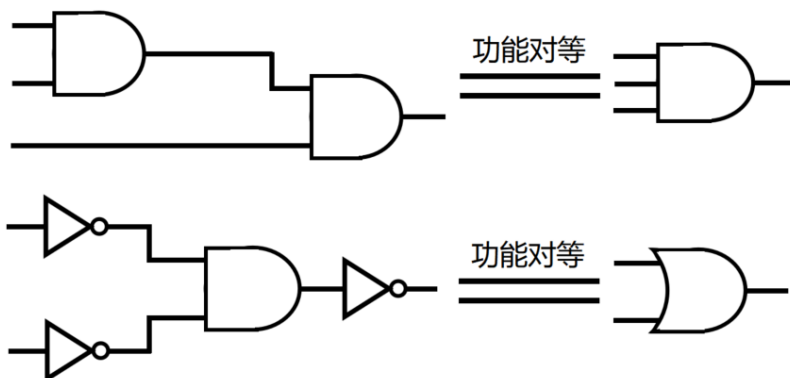
这与实验一的双控开关电路的真值表是一致的。真值表代表着一个器件输入输出所有的可能情况，所以真值表一致就意味着这两个电路的功能一致。那么，电路结构看起来很复杂的双控开关逻辑门电路，其实只需要一个异或门就可以完成任务。



所以对双控开关电路的综合过程转变为如下图所示。



上图中橙框的内容是综合中逻辑优化过程。逻辑优化将可以实现同样功能（真值表一样）的电路互相转换，已达到使用更少数量逻辑门来表示相同功能的电路的目标。类似于这样的电路还有很多，下面简单列举两个。



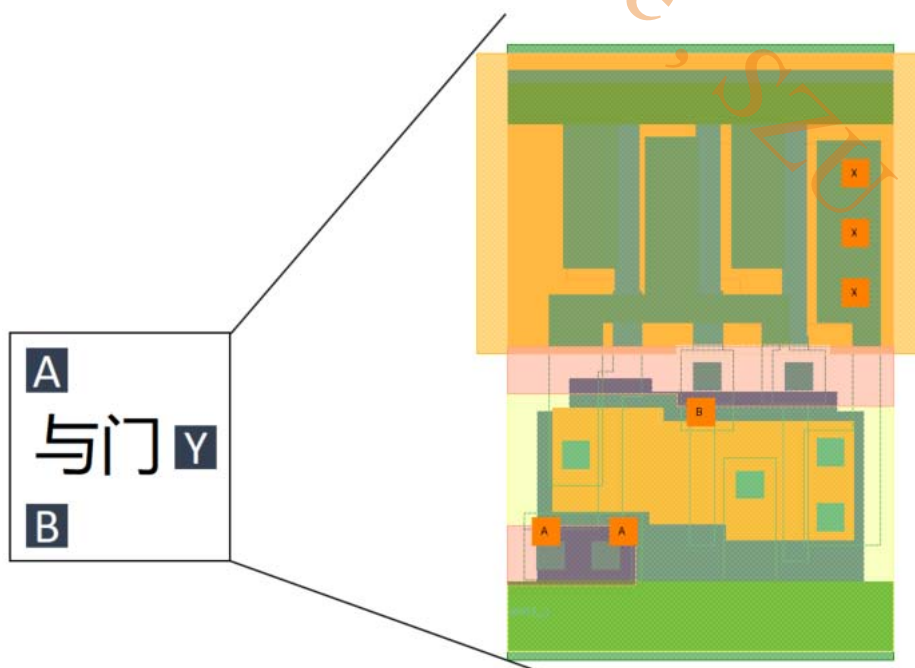
在综合中逻辑优化是最常见的优化手段。

工艺库介绍

如果把标准单元看成积木，那么工艺库就是装积木的箱子，这个箱子内有许多不同种类的积木，同理，工艺库也有许多不同种类的标准单元。

在之前介绍的那四个逻辑门（非门，与门，或门和异或门）都有对应相同功能的标准单元。并且，不同的工艺库标准单元的样子也不同。

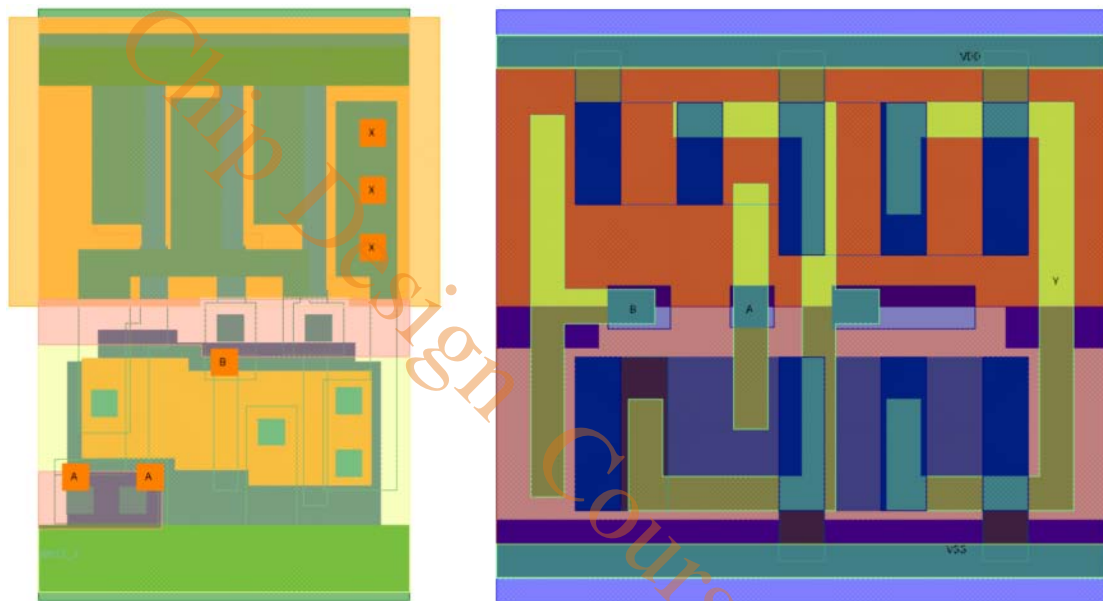
如下图所示，右图为工艺库中的与门。在右图中可以看到 6 个字母，这 6 个字母是 A、A、B、X、X、X，这些是代表标准单元的接口，其中，左图的“A”与右图的“A”对应，左图的“B”与右图的“B”对应，左图的“Y”与右图的“X”对应。在右图中有两个“A”，接入标准单元的 A 接口时，只会选择一个进行接入，同理三个“X”中也只会选择一个来接出。



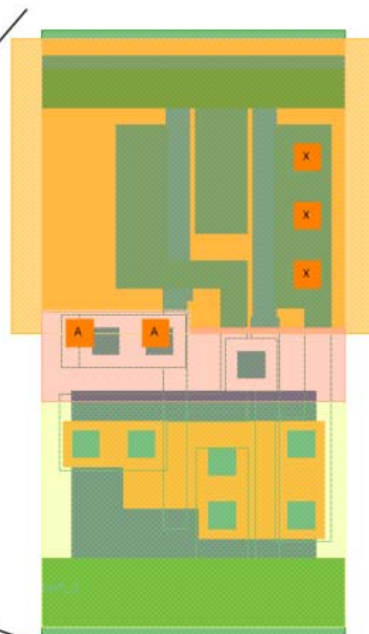
除了这些接口外，标准单元中还有很多不同颜色交叠的矩形，这些是标准单元实现逻辑的物理结构，这里只做展示，不解释其实现原理，在后续的步骤中也不会用到这些物理结构。

制造积木的厂商有很多个，每个厂商制造的积木外表风格都不一样。同样的，制造工艺库的厂商也有很多个，他们对于同一种功能的逻辑门其实现的原理不同，物理结构也不同。

如下图所示，这两个标准单元的功能一致，都是“与门”，但其接口和物理结构却有很大的差别。



除了逻辑门外，在实验一的时钟树综合阶段，时钟信号从时钟源在互连线中传递，但是过长的互连线会使得接收端收到的信号强度非常弱，低于了有效信号强度，为了保证一定的信号强度，会在互连线中间加上类似于“泵”的标准单元，被称为“缓冲器（buffer）”。



buffer 不只是用于时钟信号，时钟信号本质上也是电，只要是电信号弱，电信号弱的表现通常是很长的互连线导致的，这就可以通过在互连线中间插入 buffer 来加强信号。在电路中的信号并不是真正的 0 或 1，而是一定的电压范围，当电压值高于某个阈值时，逻辑值为 1，当低于一定的阈值时，逻辑值为 0。

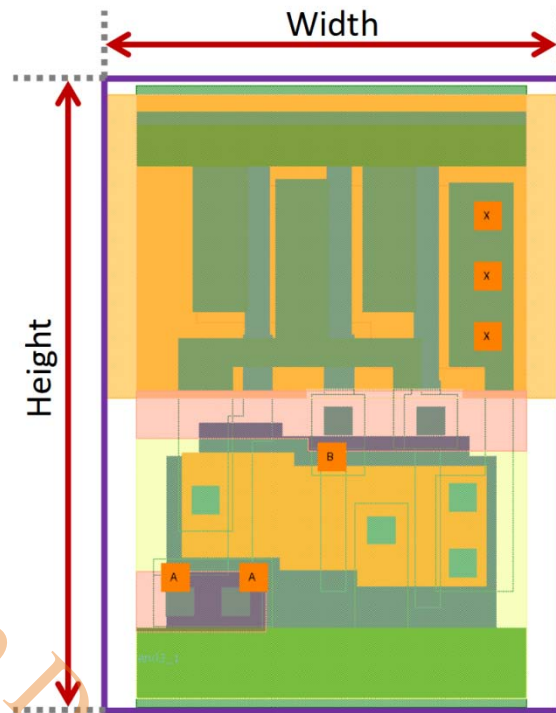
buffer 不会改变信号的逻辑值，只会使信号有更强的“辨识度”，“辨识度”指的是当信号逻辑值为 1 时，经过 buffer 会使得电压值更高，而逻辑值为 0 时，经过 buffer 会使得电压值更低。

单元置换

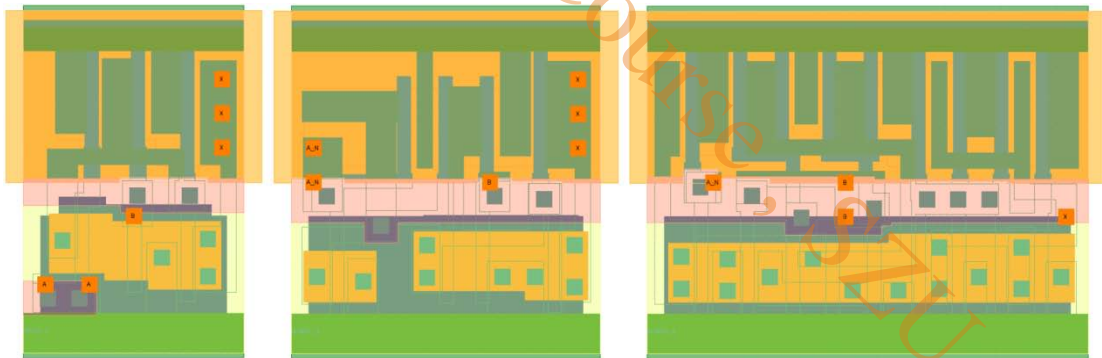
单元置换是将逻辑门转换为标准单元的过程。



单元置换有许多优化目标，包括面积、驱动能力等。这里以面积为例子，在工艺库内有多个实现相同逻辑功能的标准单元。



如下图所示，这是三个“与门”，它们都实现了相同的功能，但是面积却有很大的不同。在单元置换选择时，如果只是以面积为目标，选择面积最小为最优，虽然一个标准单元面积之间的差异并不大，但是在大规模的电路下，这是一个很客观的优化。



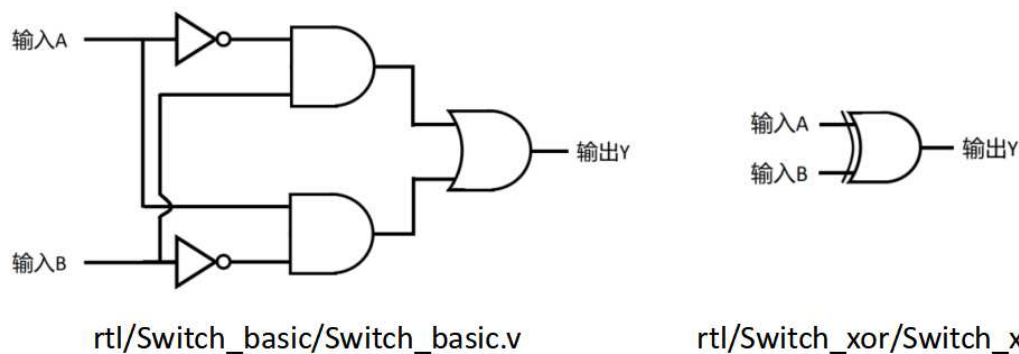
当然，单元不是越小越好，越小的单元在其他能力（如驱动能力等）上可能越差，所以用什么单元进行置换是需要得多因素下的折中选择。

实验步骤

逻辑优化

在逻辑优化这个实验下，我们提供了两个例子，这两个例子的逻辑门电路分别对应下图中的左图和右图。我们在 iFlow 中准备了对应于这两个设计的前端设计 `Switch_basic`（只由三个基础逻辑门组成的双控开关电路）和 `Switch_xor`（用

一个异或逻辑门组成的双控开关电路）。设计文件的路径在电路图下标出。



在内容部分“逻辑优化”的学习中，我们知道这两个逻辑门电路的真值表是一样的，所以它们的电路功能一样。所以，将左侧电路进行综合后，将会得到右侧电路，现在我们来实验一下。

首先在 ubuntu 桌面上打开终端，进入 iFlow 目录下。

```
lc3@lc3-virtual-machine:~/Desktop$ cd iFlow/  
lc3@lc3-virtual-machine:~/Desktop/iFlow$ pwd  
/home/lc3/Desktop/iFlow  
lc3@lc3-virtual-machine:~/Desktop/iFlow$
```

启动综合工具查看电路图，这里使用的是开源综合工具 yosys。

```
lc3@lc3-virtual-machine:~/Desktop/iFlow$ ./tools/yosys  
  
-----  
| 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.  
|  
|-----  
  
Yosys 0.9+1706 (git sha1 4be891e8, gcc 9.3.0-10ubuntu2 -fPIC -Os)  
  
yosys> |
```

首先查看 Switch_basic 设计的电路图。分别输入以下两行命令。

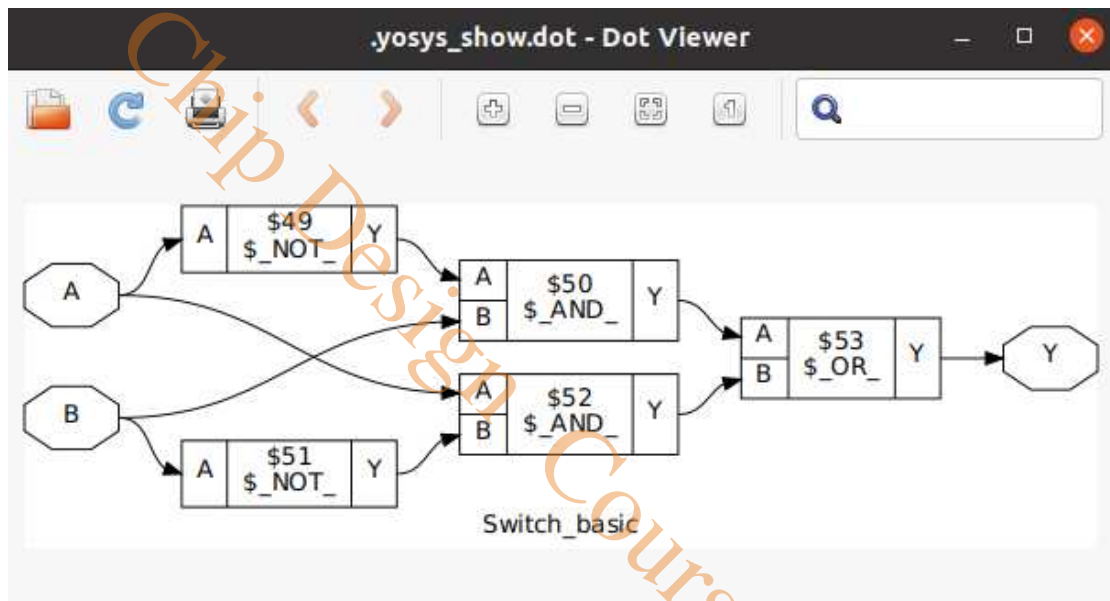

```

yosys> read_verilog rtl/Switch_basic/Switch_basic.v
1. Executing Verilog-2005 frontend: rtl/Switch_basic/Switch_basic.v
Parsing Verilog input from `rtl/Switch_basic/Switch_basic.v' to AST representation.
Generating RTLIL representation for module `Switch_basic'.
Successfully finished Verilog frontend.

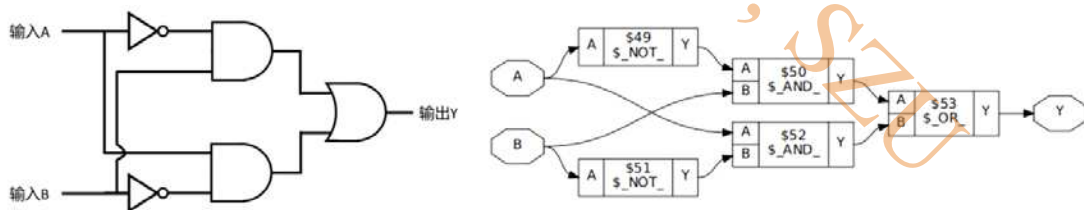
yosys> techmap; opt; show Switch_basic
2. Executing TECHMAP pass (map to technology primitives).
2.1. Executing Verilog-2005 frontend: <techmap.v>

```

在运行结束后会弹出显示前端设计对应的电路图。这是还未经过综合的电路图。



可以看到，综合工具看到的电路图和之前的逻辑电路图是一致的。



关闭图查看器，退出 yosys。

```

Exec: { test -f /home/lc3/.yosys_show.dot.pid && ( echo $$ >&3; exec
ow.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec
ot'; ) 3> '/home/lc3/.yosys_show.dot.pid' &

yosys> exit

End of script. Logfile hash: c6c040c123, CPU: user
1 MB peak
Yosys 0.9+1706 (git sha1 4be891e8, gcc 9.3.0-10ubun
Time spent: 25% 2x opt_expr (0 sec), 20% 1x show (0
lc3@lc3-virtual-machine:~/Desktop/iFlow$

```

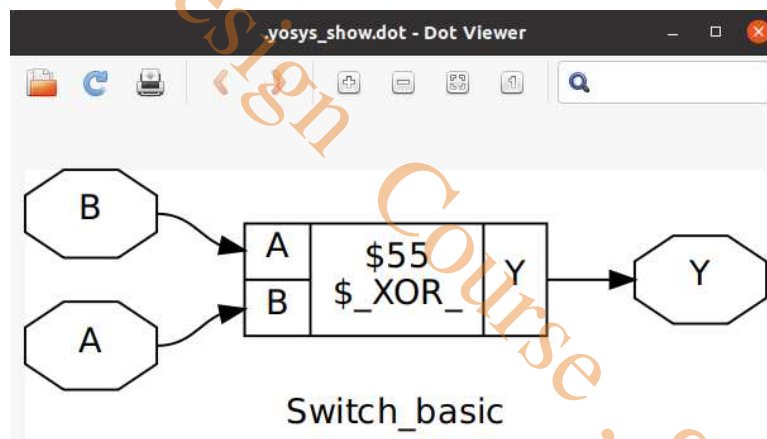
接下来我们用 iFlow 调用自动化脚本对其综合。进入 scripts 目录，并输入 iFlow 综合命令。

```

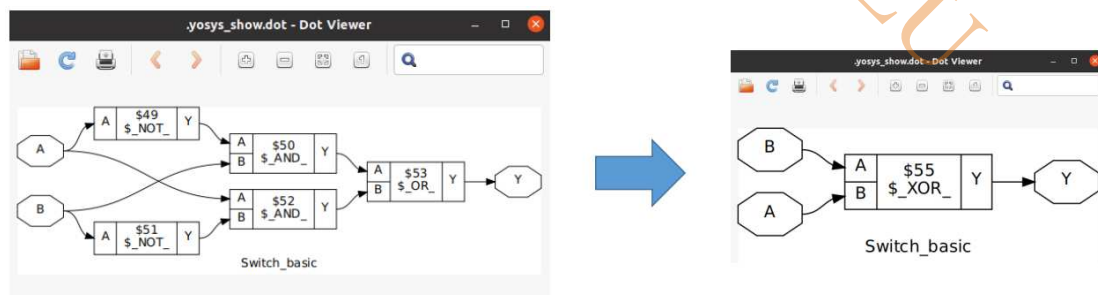
lc3@lc3-virtual-machine:~/Desktop/iFlow$ cd scripts/
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ pwd
/home/lc3/Desktop/iFlow/scripts
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ ./run_flow.py -d Switch_b
asic -s synth -f sky130 -t HS -c TYP

```

运行结束后弹出综合后的电路图。



所以通过综合，原电路图的优化过程如下。



再通过 yosys 查看 Switch_xor 的电路图。

```
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ cd ..
lc3@lc3-virtual-machine:~/Desktop/iFlow$ pwd
/home/lc3/Desktop/iFlow
lc3@lc3-virtual-machine:~/Desktop/iFlow$ ./tools/yosys

-----
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.
-----

Yosys 0.9+1706 (git sha1 4be891e8, gcc 9.3.0-10ubuntu2 -fPIC -Os)

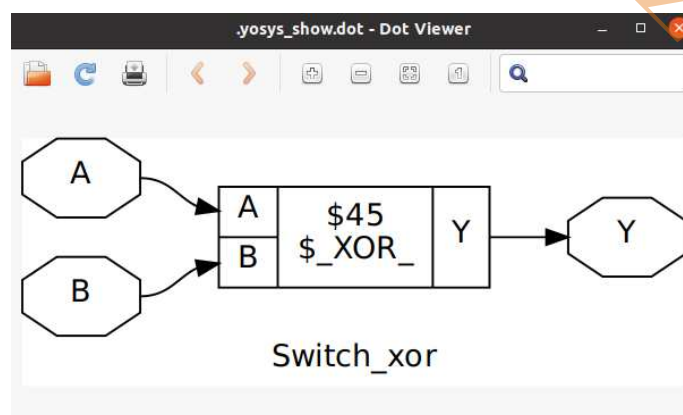
yosys> 
```

输入以下命令查看。

```
yosys> read_verilog rtl/Switch_xor/Switch_xor.v
1. Executing Verilog-2005 frontend: rtl/Switch_xor/Switch_xor.v
Parsing Verilog input from `rtl/Switch_xor/Switch_xor.v' to AST representation.
Generating RTLIL representation for module `Switch_xor'.
Successfully finished Verilog frontend.

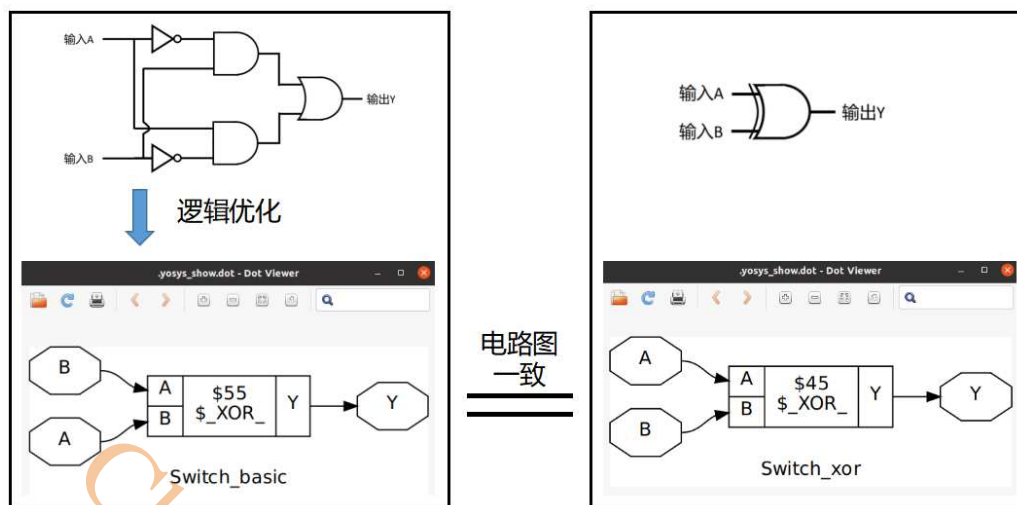
yosys> techmap; opt; show Switch_xor
2. Executing TECHMAP pass (map to technology primitives).
```

Switch_xor 的逻辑电路图如下。



所以由以上实验可以看出，将 Switch_basic 进行逻辑优化后的电路图与 Switch_xor 的电路图设计是一致的。同时说明了在综合阶段会对相同功能的电路

进行逻辑优化。



最后关闭图查看器，退出 yosys，本实验结束。

查看工艺库

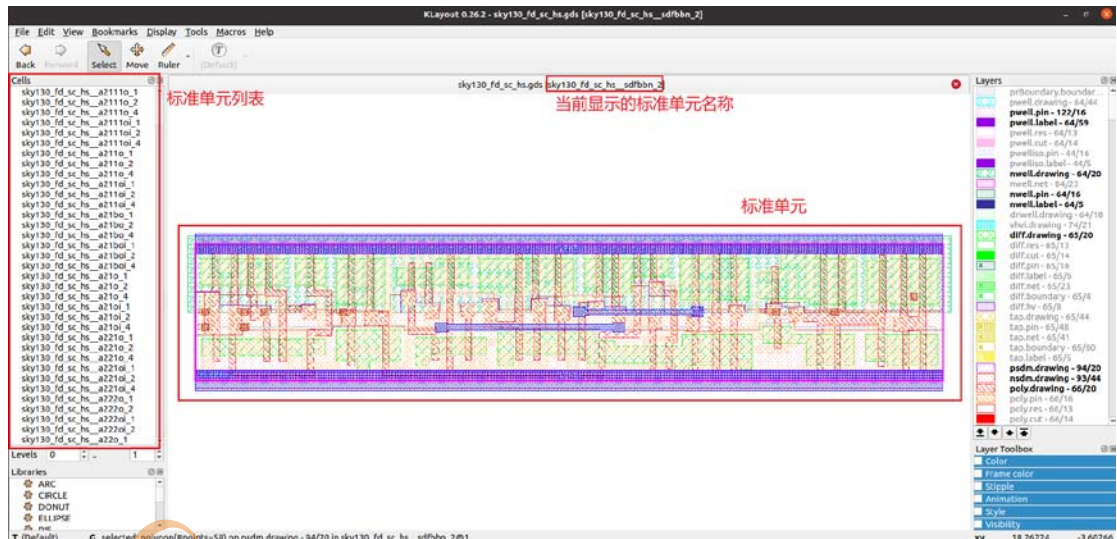
首先进入 iFlow 目录。

```
lc3@lc3-virtual-machine:~/Desktop$ cd iFlow/  
lc3@lc3-virtual-machine:~/Desktop/iFlow$ pwd  
/home/lc3/Desktop/iFlow  
lc3@lc3-virtual-machine:~/Desktop/iFlow$
```

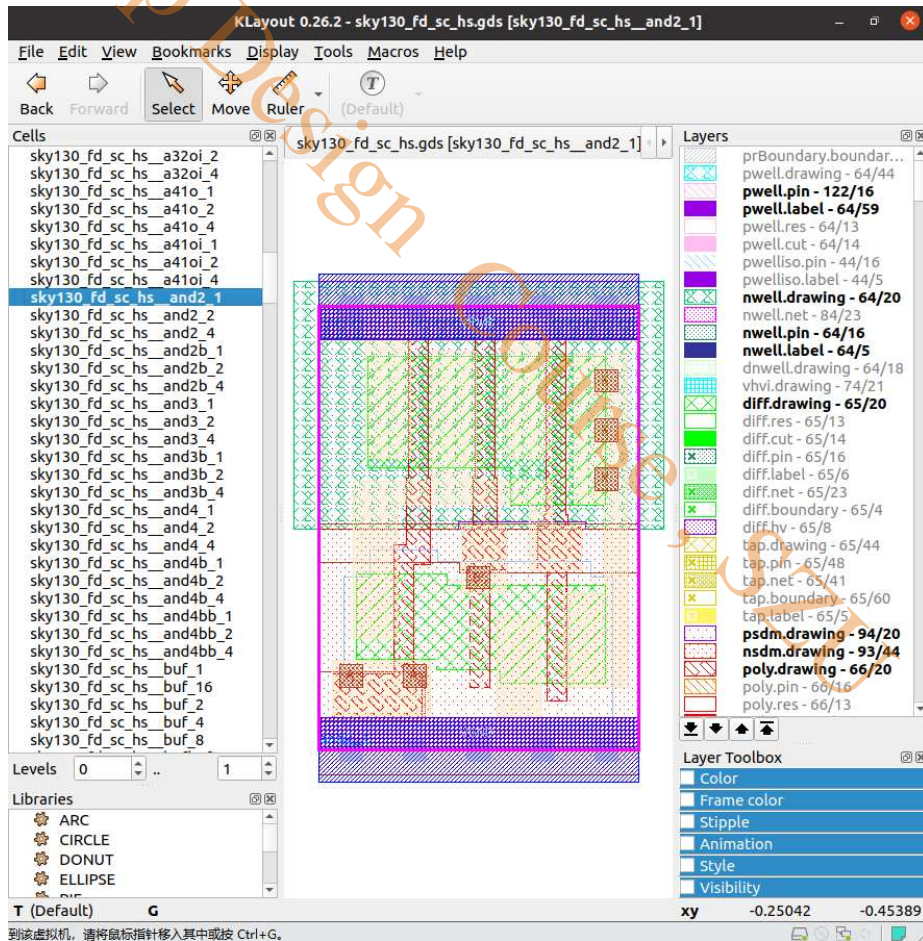
使用 klayout 工具查看工艺库内的标准单元。我们这里查看的工艺库是 skywater，这是一个谷歌的开源工艺库，130nm 工艺，简称为 sky130。

```
lc3@lc3-virtual-machine:~/Desktop/iFlow$ klayout foundry/sky130/gds/sk  
y130_fd_sc_hs.gds -l foundry/sky130/klayout.lyp
```

打开 klayout 后，左侧栏是所有标准单元的名称，中间是标准单元的“模样”，只需要对标准单元有个大致的认识即可，不需要明白其原理。



在“标准单元列表”内找到名称为“sky130_fd_sc_hs_and2_1”，对其右键并选择“Show As New Top”，在中间就会显示此标准单元的样子。

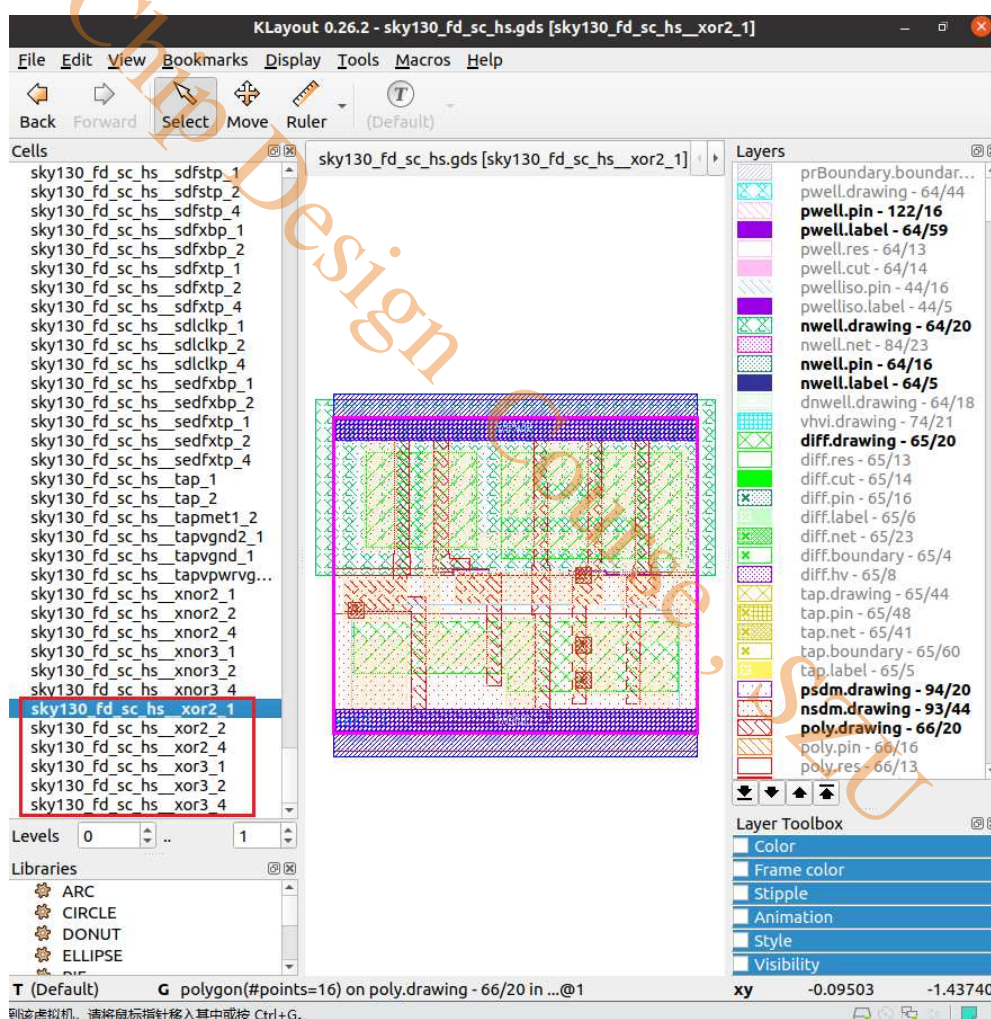


这个标准单元“sky130_fd_sc_hs_and2_1”实现的功能是两输入与门。在标准单元列表内还有很多以“sky130_fd_sc_hs_and”开头的单元名称，这些都是不同类型的与门（还有三输入，四输入的）。只要对名称右键后选择“Show As New

Top” 就可以查看对应标准单元。

```
sky130_fd_sc_hs_and2_1
sky130_fd_sc_hs_and2_2
sky130_fd_sc_hs_and2_4
sky130_fd_sc_hs_and2b_1
sky130_fd_sc_hs_and2b_2
sky130_fd_sc_hs_and2b_4
sky130_fd_sc_hs_and3_1
sky130_fd_sc_hs_and3_2
sky130_fd_sc_hs_and3_4
sky130_fd_sc_hs_and3b_1
sky130_fd_sc_hs_and3b_2
sky130_fd_sc_hs_and3b_4
sky130_fd_sc_hs_and4_1
```

将标准单元列表滑动到最底部，可以看见几个异或门标准单元。



如图所示的就是二输入的异或标准单元。在之后的学习里，经常能看到标准单元，无论是从形状上，还是名称上，只需要能认出是标准单元即可，不要求知道其物理实现原理。

不同工艺库下的单元置换

进入 iFlow/scripts 目录下。

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

在这里还是使用 Switch_basic 设计,对这个设计进行综合。在这一行命令中,红框内是使用工艺库“sky130”进行综合。

```
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ ./run_flow.py -d Switch_basic -s synth -f sky130 -t HS -c TYP
```

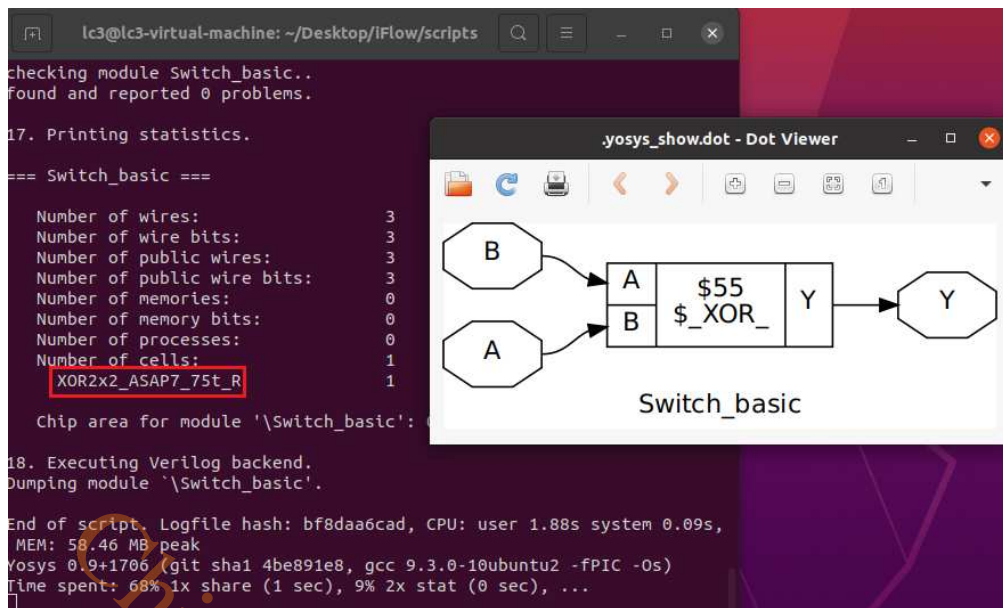
运行命令,命令运行结束后可以看到,综合将 Switch_basic 优化为了只有一个异或门,并且使用了 sky130 工艺库内的 sky130_fd_sc_hs__xor2_1 这个异或标准单元。

The image shows a terminal window and a Yosys Show Dot Viewer window. The terminal window displays the output of the synthesis command, including statistics for the Switch_basic module. The Yosys Show Dot Viewer window shows a circuit diagram of the Switch_basic module, which is an XOR gate. The inputs are labeled A and B, and the output is labeled Y. The gate is labeled sky130_fd_sc_hs__xor2_1.

sky130_fd_sc_hs__xor2_1 这个异或标准单元就是“查看工艺库”实验最后的那个单元。关闭“Dot Viewer”，接下来我们将工艺库换为 asap7 进行综合。

```
lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ ./run_flow.py -d Switch_basic -s synth -f asap7 -t HS -c TYP
```

命令运行结束后,在“statistics (统计)”中可以看到使用的不再是 sky130 工艺库的异或标准单元了,而是 asap7 的异或标准单元。



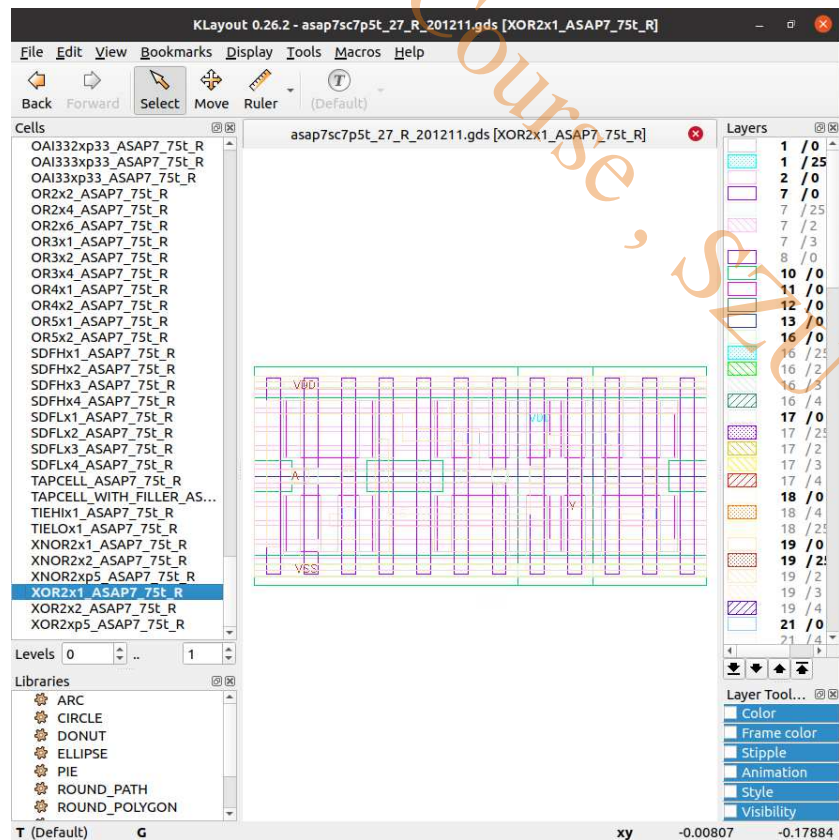
关闭“Dot Viewer”，返回 iFlow 目录下，输入命令打开 asap7 工艺库。

```

lc3@lc3-virtual-machine:~/Desktop/iFlow/scripts$ cd .. 返回上一级
lc3@lc3-virtual-machine:~/Desktop/iFlow$ pwd
/home/lc3/Desktop/iFlow
lc3@lc3-virtual-machine:~/Desktop/iFlow$ klayout foundry/asap7/gds/asap7sc7p5t_27_R_201211.gds -l foundry/asap7/klayout.lyp

```

找到并查看对应的异或标准单元。



查看网表

网表是综合的输出结果，这里将学习如何查看综合输出的网表文件。

进入 iFlow 目录下，通过“cat”命令浏览文件内容。

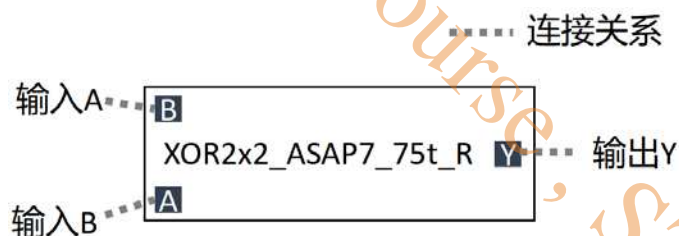
```
lc3@lc3-virtual-machine:~/Desktop$ cd iFlow/
lc3@lc3-virtual-machine:~/Desktop/iFlow$ pwd
/home/lc3/Desktop/iFlow
lc3@lc3-virtual-machine:~/Desktop/iFlow$ cat result/Switch_basic.synth.yosys_0.9
.HS.TYP.default/Switch_basic.v
/* Generated by Yosys 0.9+1706 (git sha1 4be891e8, gcc 9.3.0-10ubuntu2 -fPIC -Os
) */

module Switch_basic(A, B, Y);
input A;
input B;
output Y;
XOR2x2_ASAP7_75t_R _0_ (
    .A(B),
    .B(A),
    .Y(Y)
);
endmodule
lc3@lc3-virtual-machine:~/Desktop/iFlow$
```

网表

存储结果文件的文件夹名称规则是{设计名.阶段名*****}，所以这次我们打开的是 Switch_basic 设计在 synth（综合）阶段的结果。

从网表可以看出，其连接关系是这样的。



输入 A 与单元的 B 接口连接，输入 B 与单元的 A 接口连接，实际上，字母颠倒没有什么问题，因为只是一个代号，功能还是一样的。

小结

综合的主要过程是，先将硬件描述语言转换为逻辑门电路，再将逻辑门电路转换为标准单元之间具有连接关系的电路图（网表，netlist）。

在实验二综合中，我们学习了电路的逻辑优化是如何完成的，了解了电路优化的基本概念。学习了如何查看工艺库内的某个标准单元。并且使用了不同工艺库对同一个设计进行逻辑综合。还学会了查看综合的结果网表。

作业

- (1) 查看 ALU 综合前后电路图，ALU 的设计文件路径是 “rtl/ALU/ALU.v”。
- (2) 查看 sky130 工艺库内的 “sky130_fd_sc_hs__or2_1” 单元并截屏。
- (3) 使用 sky130 和 asap7 工艺库查看 ALU 的统计结果(如下图所示)。

