



集成电路设计实习 VLSI Design Labs

## 单元实验四

数字系统设计-前端

## 实验目的及时间安排

---

- 掌握数字系统的半定制设计方法
- 完成16位加法器的RTL级电路设计和仿真
- 完成16位加法器的门级电路设计和仿真
- 完成逻辑综合
- 设计时间：1次课
- 设计数据和上机指导在lab4.tar文件中

## 内容安排

---

- 前端设计：RTL逻辑仿真、逻辑综合、门级网表仿真
- 后端设计：自动布局布线和版图验证
- 标准单元库：SMIC 0.35um工艺库，库文件已经放在实验的lib文件夹中，为数字半定制设计中使用的SMIC 0.35um工艺标准单元库，包括支持逻辑仿真、逻辑综合、自动布局布线和版图验证的设计文件
- 本实验使用Cadence公司的EDA工具完成

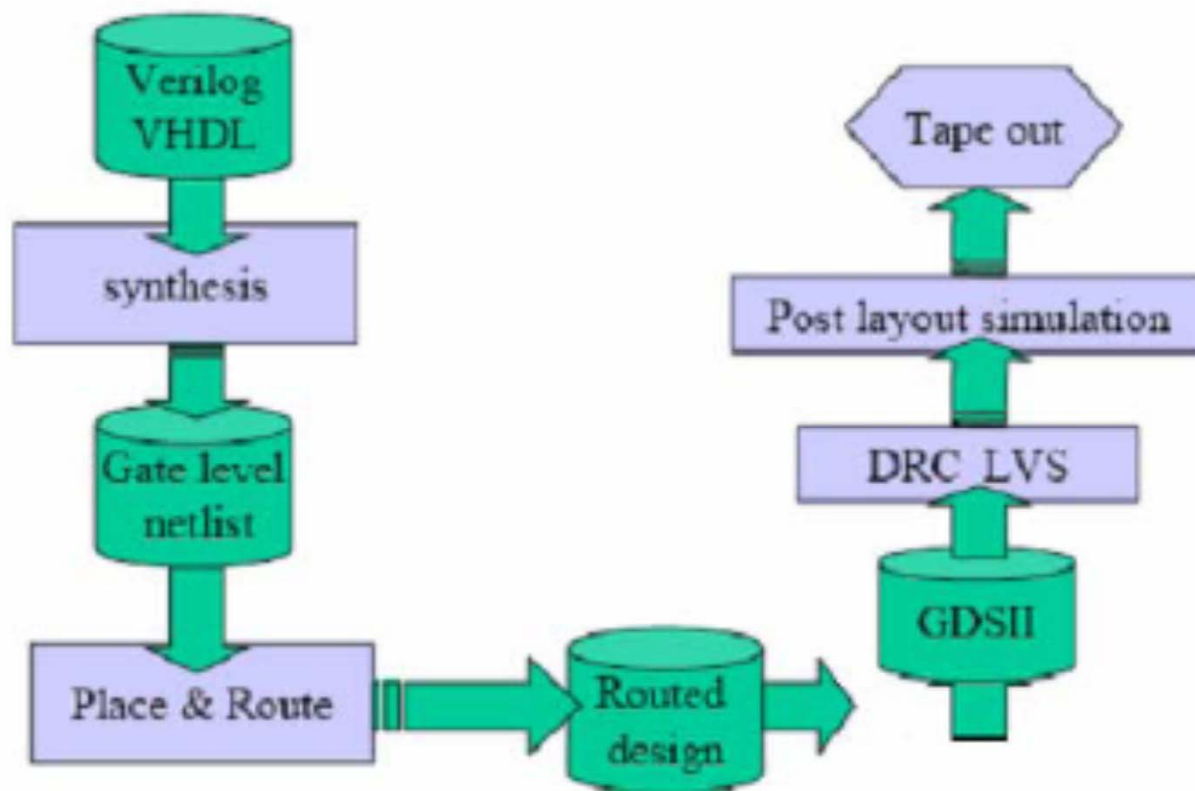
## 实验过程

---

- 在登陆根目录下解压lab4.tar文件，进入产生的lab4目录，分别在仿真sim、综合syn、布局布线layout和版图验证lvs目录下完成本单元的实验内容，其余文件夹为仿真、综合相应库文件目录
- 分别完成16位加法器的RTL级和门级设计，理解不同设计方法的特点
- 对完成的设计进行逻辑仿真验证功能的正确性
- 对验证正确的设计进行逻辑综合，观察不同约束条件下，综合器生成的不同设计
- 对综合后的门级网表进行逻辑仿真，验证其正确性，并观察其门级时序
- 输出RTL设计的门级网表用于后端设计

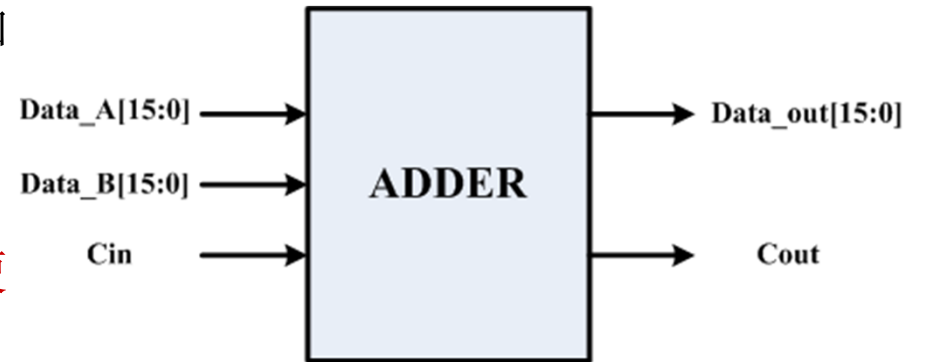
# Cell-based ASIC 设计流程

- 基于标准单元的半定制设计流程



# 前端设计1——16bit加法器的RTL设计

- 设计要求：电路完成带进位的2个16位二进制数的加法操作，输出16位的‘和信号’以及1位的‘进位输出信号’
- 端口定义如下表
- 用硬件描述语言进行设计输入，**推荐使用verilog语言**
- 完成RTL级仿真，逻辑综合和门级仿真



端口名	位宽: 单位 bit	说明
<u>Data_a</u>	16	加法器数据输入
<u>Data_b</u>	16	加法器数据输入
<u>Cin</u>	1	加法器进位输入
<u>Data_out</u>	16	加法器数据输出
<u>Cout</u>	1	加法器进位输出

## 前端设计1 ——文件管理

---

### ●lab4

- src: verilog源文件
- sim: 仿真工作目录
- syn: 逻辑综合目录
- layout: 后端版图工作目录
- netlist: 综合得到的网表文件
- lib: 库文件目录
- lvs: 版图验证目录
- lef: 后端库目录

## RTL设计——第一步：逻辑仿真

---

- 在根目录下键入 `> cds3`
- 进入 `lab4/src` 目录
- 在 `src` 目录下，编写16位加法器设计文件 `adder.v` 以及测试文件 `adder_tb.v`，本实验测试文件已经给出，不需要编写了
- 进入 `lab4/sim` 目录，编写 `run.f`，内容如下：

```
../src/adder.v  
../src/adder_tb.v  
-y 和 -v 命令是包含单元库的命令
```

本实验已经给出 `run.f` 文件，大家只需要打开看看，  
读懂里面命令的意思



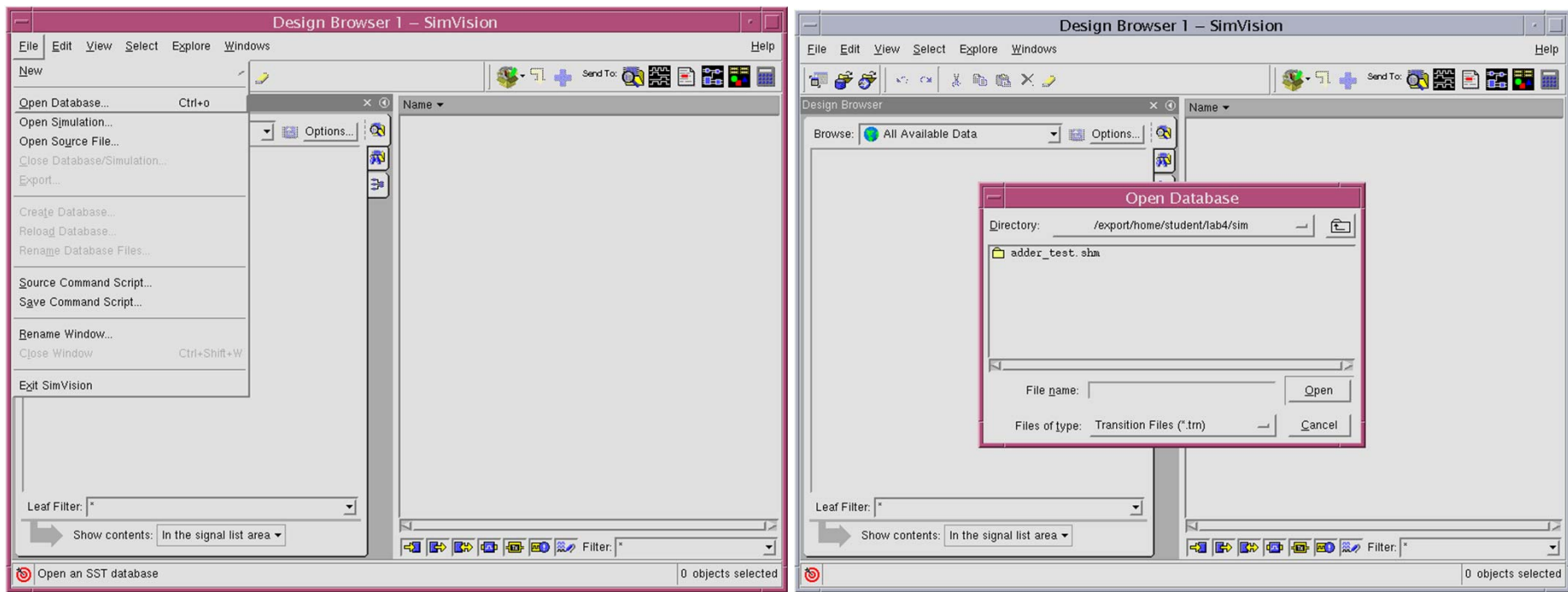
## RTL设计——第一步：逻辑仿真

---

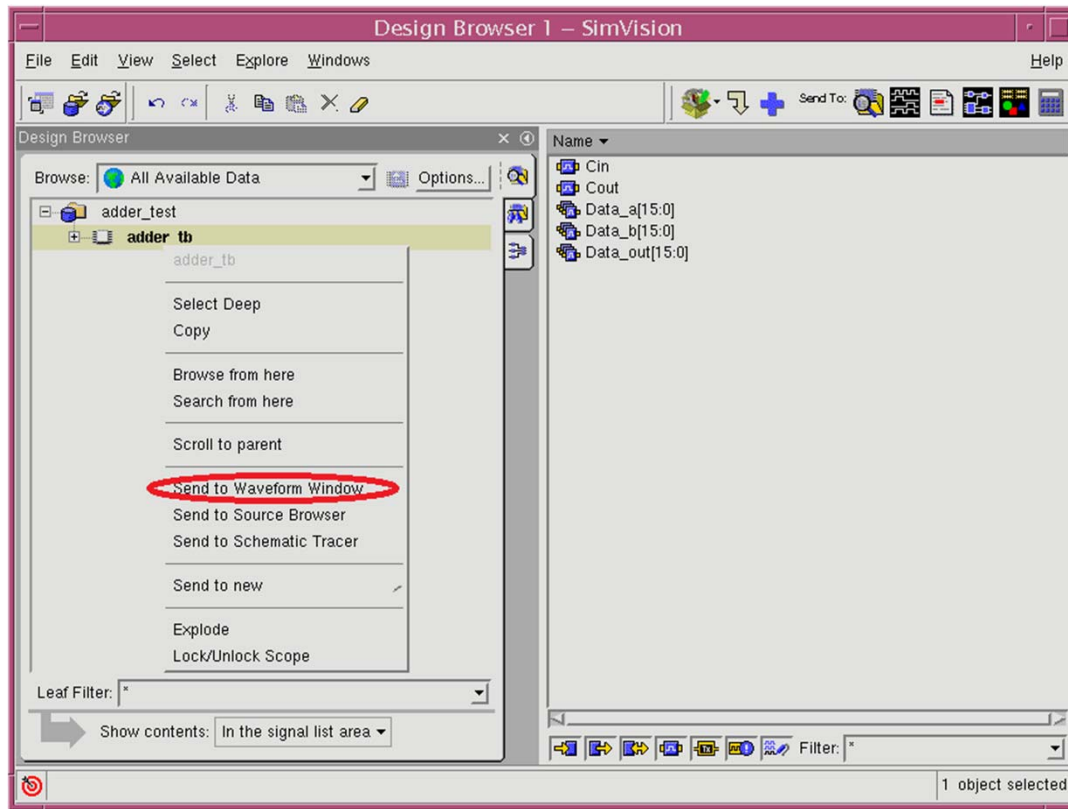
- 对完成的verilog设计进行逻辑仿真，验证语法及设计的逻辑功能
- 使用Cadence公司的VerilogXL仿真器
- 在当前目录下，在命令窗口启动Verilog-XL仿真器，进行仿真
- 输入 `verilog -f run.f` 命令

# RTL设计——第一步：波形观察

- 观察波形需要在测试文件中开启波形记录的命令，大家可以打开测试文件看看：**\$shm\_open**
- 目录：在当前目录lab4/sim下
- 命令行键入：**simvision &** 命令，启动波形观察工具

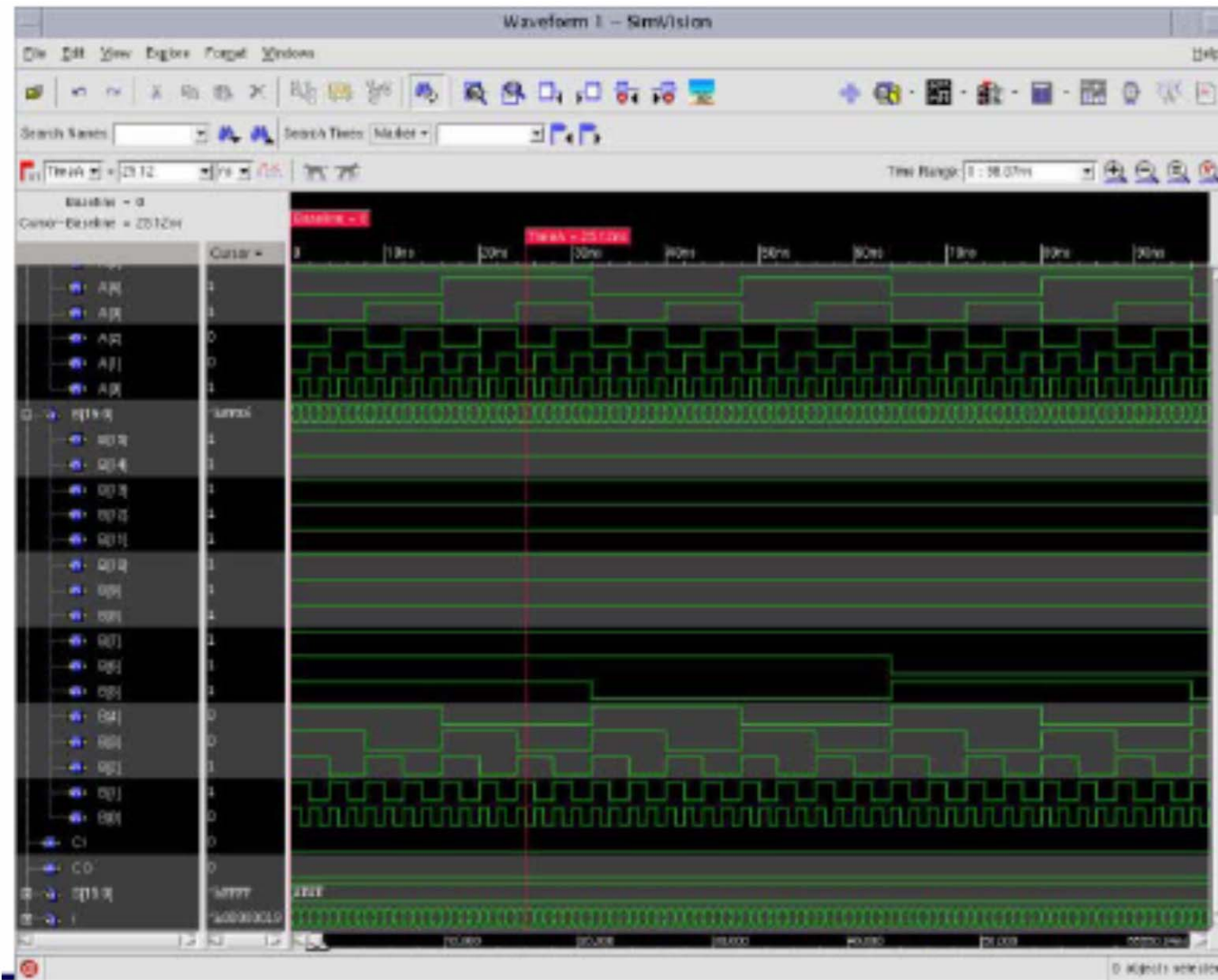


# RTL设计——第一步：波形观察



- 观察波形，验证仿真结果，通过波形检查器检测逻辑结果

# 通过波形检查逻辑功能



## RTL设计——第二步：逻辑综合

---

- 综合单元库： simc 0.35um工艺
- 对于逻辑仿真验证正确的RTL级加法器代码，进行逻辑综合，将其映射到标准单元库，生成门级网表
- 重新打开一个命令窗口，在根目录下键入 `> soc52`
- 进入lab4/syn目录，其中的my.tcl文件为综合命令的脚本文件，本实验已经给出，大家打开阅读，可以根据需要修改；
- smic35os142\_typ.lib文件（已经放置到实验目录的lib文件夹中）是单元库的逻辑综合模型，3.3v电压，typical工艺corner；
- 在syn路径下，键入 `>rc` 命令启动RTL Compiler综合器；
- 在命令窗口中输入 `> source my.tcl` 启动其中的综合命令。
- 下面给出my.tcl中综合命令的说明：

## 逻辑综合：读入单元库

---

- Shell命令中使用的是tcl语法，根据提供的my.tcl中的例子，可以写出自己的tcl命令
- 利用set\_attribute lib\_search\_path <full\_path\_of\_technology\_library\_directory> / 指定综合用的库文件目录。
- 利用set\_attribute library <technology\_library> 来指定综合的库文件
- （利用文本编辑器打开库文件lib，观察文件中的内容，了解库中的单元种类，熟悉综合库的结构）
- 逻辑综合就是要把RTL设计映射到该标准单元库中

## 逻辑综合：读入RTL设计

---

- 利用`set_attribute hdl_search_path`  
`<full_path_of_technology_library_directory>` / 指定设计的源代码文件目录。
- 利用`read_hdl <hdl_file_names>`读入16位加法器的RTL设计文件。
- 进行时序约束之前用`elaborate <top_level_design_name>`将顶层设计转换为与工艺无关的逻辑拓扑。

## 逻辑综合：时序约束

---

- 参考tcl文件，对加法器设计施加约束条件，以便在逻辑综合过程中产生满足设计要求的门级设计
- 可以先将16位加法器的时序约束设为15ns，即要求逻辑综合生成一个在15ns内可以完成16位加法的设计。需要添加虚拟时钟：  
`define_clock -name clk -period 15000`（系统默认的时间单位为ps）
- （时钟周期减去input delay和output delay就是要求综合器完成的设计的关键路径延迟时间约束）

定义输入输出外部延时：

```
external_delay -clock clk -input <specify_input_external_delay_on_clock>
```

```
external_delay -clock clk -output <specify_output_external_delay_on_clock>
```



## 逻辑综合：综合

---

- 采用命令：`synthesize - to_mapped` 进行综合。
- 综合成功后需要查看综合的结果报告。

## 逻辑综合：综合结果报告

---

- 逻辑综合完成后，采用tcl命令，生成综合器对完成的设计报告。
- 报告包括时序，面积等，重点观察时序报告，考察生成的设计的速度

- 采用命令：

```
report timing > <specify_timing_report_file_name>
```

```
report area > <specify_area_report_file_name>
```

生成时序及面积报告。

- 察看设计的关键路径

## 逻辑综合：严格约束

---

- 从时序报告中可以发现，由于施加的设计约束时钟周期比较宽松，为15ns，生成的设计有较大的富余，即报告中slack为正
- 重新进行构建设计，收紧设计约束条件中的时钟周期，进行逻辑优化，可以发现随着时序约束的严格，综合器进行优化的时间增加
- 察看设计报告，观察时序报告中的slack情况，并观察面积较之前的变化
- 通常情况下，为了获得更快的设计，需要更大的面积
- 直到slack为负值，说明生成的设计已经达到优化的极限

## 逻辑综合：输出网表

---

- 采用命令

`write -mapped > <specify_netlist_name>` 导出网表。

- 观察生成的网表，可以发现由于RTL Compiler综合器中采用的是串行加法器的算法，综合后的设计为一个16位的串行加法器
- 另外`write_sdc > <specify_constraint_name>`命令是到处约束条件，在后端版图时用到

## RTL设计——第三步：门级网表仿真

---

- 对逻辑综合器生成的门级16位加法器的网表进行逻辑仿真
- 使用RTL仿真中相同的testbench文件对门级网表进行门级仿真（注意要修改run.f中的读入verilog文件的命令）
- 仿真过程中需要用到标准单元库中的仿真模型，在run.f文件中指定（-y和-v命令，已经给出）
- 比较门级仿真和RTL级仿真的波形的变化
- 采用零延迟和单位延迟（run.f中的+delay\_mode\_unit命令，去掉前面的注释符号来开启）模式分别进行仿真，并通过观察仿真结果

## 前端设计2——16位加法器的门级设计

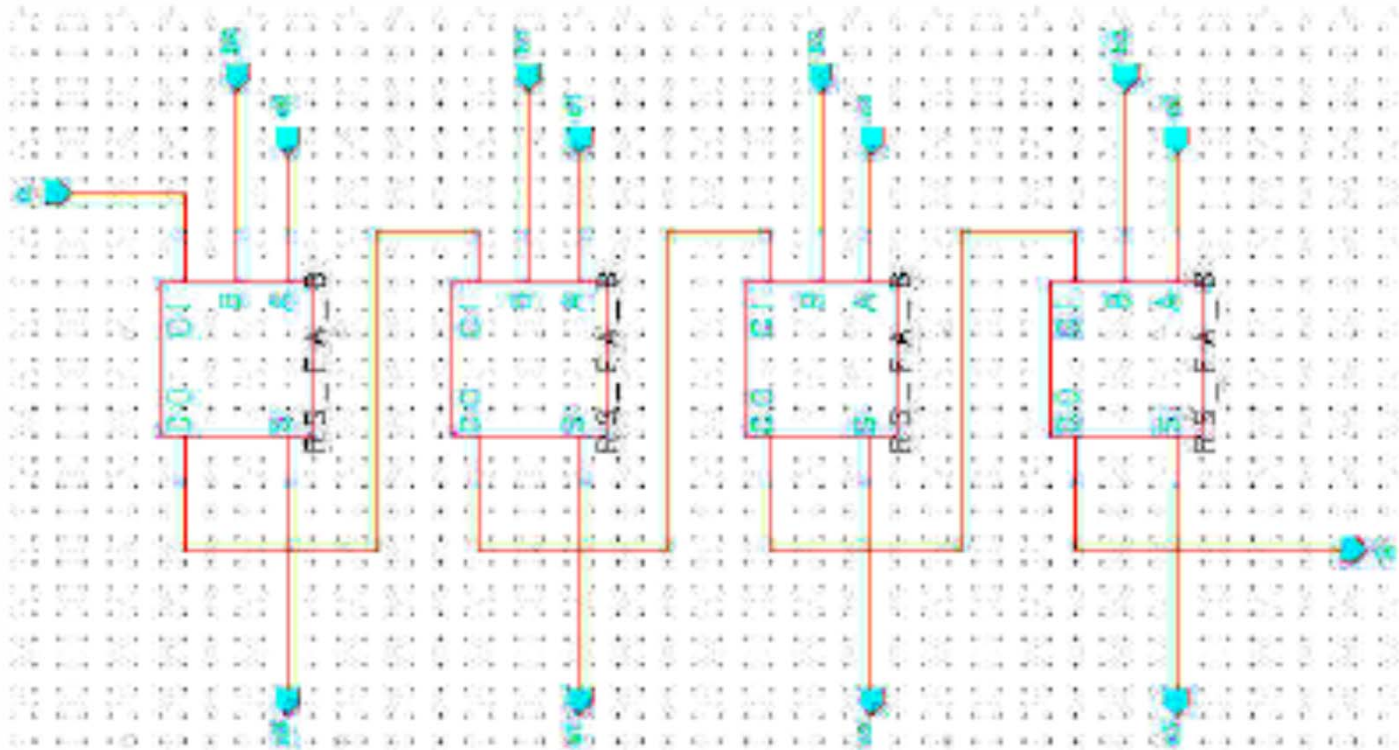
---

- 设计要求：同RTL设计要求
- 利用标准单元库中的1bit全加器单元ad01d1构建一个16位加法器的门级设计
- 库文件路径：  
/net/server1/software/smic/smic35/snpsLibs/smic35/v1.0/verilog/smic35os142/zero//ad01d1.v
- 端口：module ad01d1(A, B, CI, S, CO)
- 写代码时直接调用该模块，仿真时run.f中的-y命令已经包含该库文件，不用另外读入
- 加法器采用串行结构
- 完成门级仿真，逻辑综合

## 门级设计——第一步：设计输入

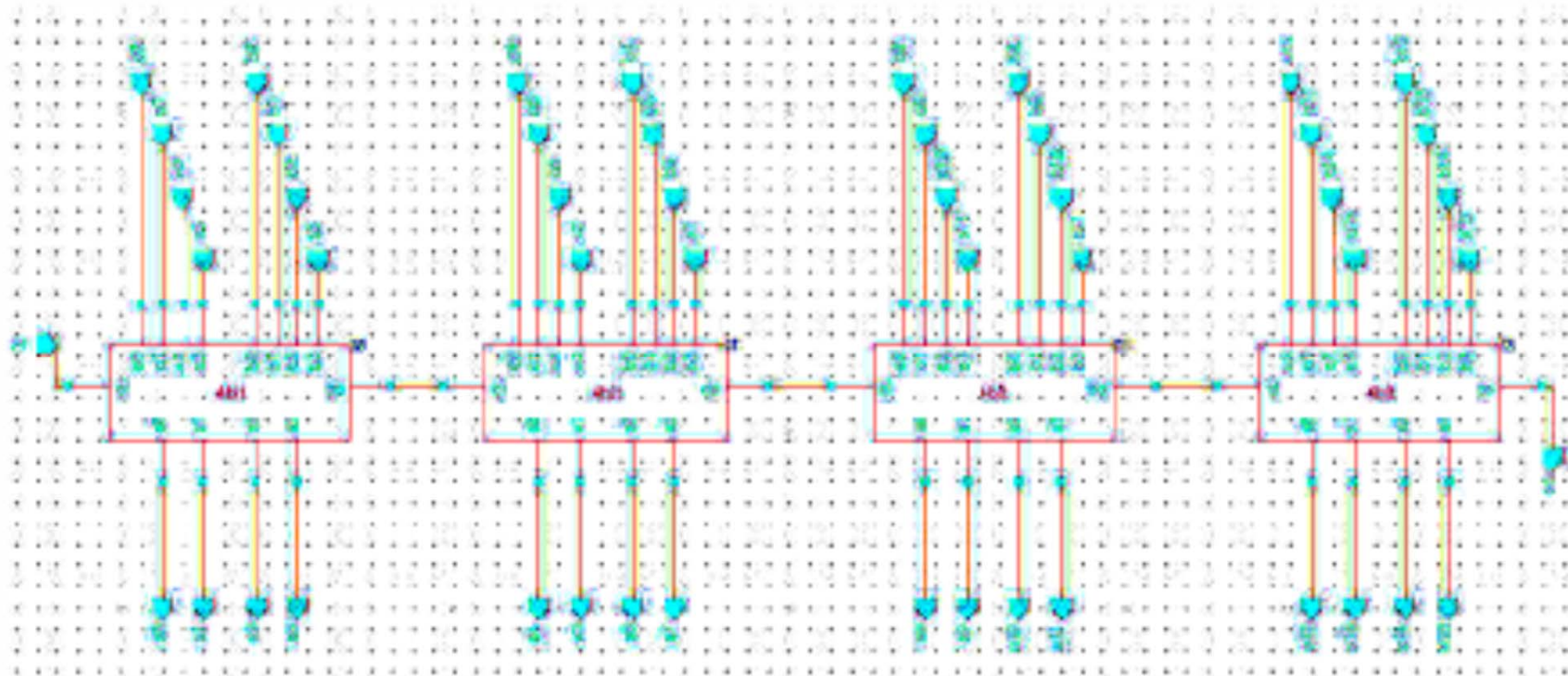
---

- 调用标准单元库中的加法器，构建4bit加法器



# 16位加法器

- 实例化4bit加法器，组成16bit串行加法器





## 门级设计——第二步：逻辑仿真

---

- 将完成的加法器的门级设计，利用前面RTL设计中的testbench，结合仿真库文件，进行逻辑仿真
  
- 观察门级电路的延迟情况

## 门级设计——第三步：逻辑综合

---

- 将完成的加法器的门级设计，读入综合器，进行逻辑综合
- 该过程的目的是通过综合器观察准确的延迟信息，以及观察综合器对层次化的设计的优化情况
- 首先将时序约束设为15ns，进行逻辑综合，观察生成的设计的时序报告，以及关键路径延迟，可以清楚的看出其关键路径是由串行的进位链组成
- 比较综合器报出的延迟信息同仿真的延迟结果，解释哪个更加准确
- 观察面积报告中的信息，并在lib文件中查到单元的面积信息，看到二者之间的关系

## 逻辑综合

---

- 将时序约束收紧，重新进行逻辑综合，观察生成的设计的时序报告，以及关键路径延迟，并观察4个串行4位加法器内部的电路结构
- 可以看出，由于时序约束收紧，综合器对4位加法器进行了逻辑优化，使用了并行加法算法
- 4个顶层加法器的串连结构并未改变
- 观察时序报告，面积报告，并同之前的综合结果进行比较

## 半定制设计总结——16比特加法器前端设计

---

- 设计要求
- 端口定义、功能分析
- 基于HDL的RTL级和门级电路设计
- 功能仿真
- 逻辑综合——将HDL代码转换为基于标准单元库的门级网表
- 综合后仿真