



集成电路设计实习 VLSI Design Labs

单元实验二

基于逻辑门的定制设计-1位全加器的设计

实验目的及时间安排

- 在实验1中我们学习了利用cadence的定制设计工具进行晶体管级的电路设计，包括原理图输入（schematic editor）、电路仿真（ADE）和定制版图设计（virtuoso XL）
- 在实验2中我们学习基于门级电路的定制设计方法，即利用基本门构建更大的电路
- 门级定制设计的目的是利用现有的设计资源快速构建原型设计，评估设计方案，后续可以再通过器件级设计进一步优化；其设计过程和工具的使用同器件级设计基本相同，只是设计起点是门电路
- 完成全加器的设计
- 设计时间：1次课

单元实验二：实验要求

- 电路功能正确，满足真值表
- 完成电路版图设计，版图验证和电路提取后仿真等设计过程
- 按照标准单元的方法摆放基本单元版图，保持单元等高相邻接
- 版图放在第一象限，最左面单元的左下角位于坐标原点
- 工艺提供3层金属，设计中可以使用金属1和金属2作为基本单元之间的互连（金属3留给顶层设计）

单元实验二：实验内容

●用基本单元设计全加器，即一位加法器

●输入信号： C_i ， A ， B ，其中 C_i 为进位输入信号， A ， B 为本位的输入信号，即被加数和加数

●输出信号： C_o ， S ，其中 C_o 为本位的进位输出信号， S 为本位的和信号

●真值表：根据真值表可以得到全加器的逻辑功能

A	B	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = A \oplus B \oplus C_i$$

$$C_o = AB + BC_i + C_iA$$

全加器:

- 用基本单元设计全加器，即一位加法器
- 根据真值表得到的逻辑表达式中需要异或和与或等逻辑功能，可以把这些逻辑功能用一个门电路实现，就是晶体管级的设计，但是由于器件较多，这个设计比较复杂，我们希望利用提供的单元库中的门电路构建加法器的版图
- 设计资源：**Chart**标准单元库，本实验可利用其库单元的版图构建定制加法器
- chrt35dg_COREcell**下的**doc**目录下有库的相关文档的说明，其中有所有单元的逻辑功能描述等详细的信息。我们在构建全加器的过程中可以参考说明文档选择需要的标准单元
- 单元的名称：从单元的名称中我们可以大体了解其基本功能，例如与、或、异或等，其名称中的**DX**表示该单元的驱动能力

chrt35dg_COREcell库的目录

- chrt35dg_COREcell目录中为数字半定制设计用标准单元库文件，里面各子目录内容如下所述：
 - **cdl**目录中为标准单元和IO单元的电路网表，用于**dracula**工具的**lvs**检查
 - **spice**目录中为单元的**spice**网表
 - **doc**目录中为标准单元和IO单元的说明文档
 - **dracula**目录中为**dracula**工具的**drc**和**lvs**检查文件，针对我们的设计进行了部分修改
 - **gds**目录中为标准单元和IO单元的版图，为**gds**格式
 - **icons**目录中为逻辑综合中单元的**symbol**，可以不用
 - **Se**目录中为**cadence**后端设计用数据
 - **Starrc**目录中为物理设计中互连线的**RC**提取文件
 - **synopsys**目录中为逻辑综合文件
 - **verilog**和**vhdl**目录中为库的逻辑仿真模型，我们使用**verilog**模型
 - **Virtuoso**目录中为单元库提供的版图设计工艺和显示文件，我们使用模拟库中的文件即可

实验步骤1——建立设计库

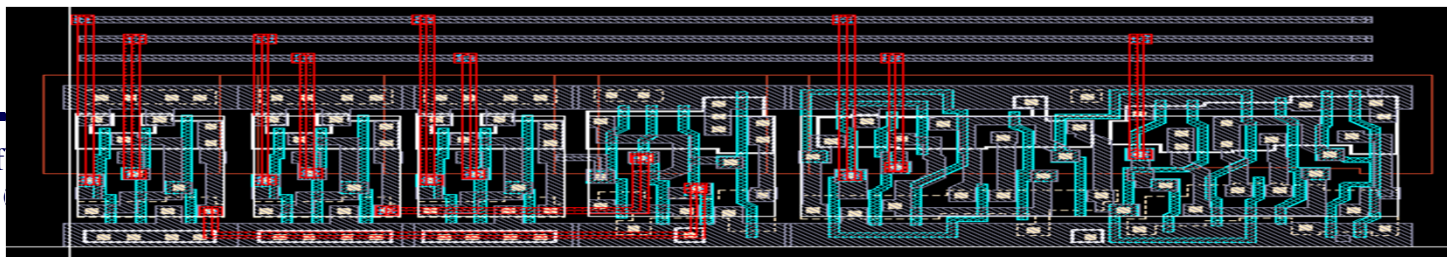
- 在lab2下的terminal中启动dfll工具（也称作virtuoso平台），即输入icfb&命令
- 利用library manager工具可以看到你在cds.lib文件中include进来的库，即是lab1中的库，如果没有看到这些库，检查cds.lib文件是否正确
- 这些库中可以看到你在lab1中实验的设计库，你可以把lab2部分的实验内容也放在这个设计库中，当然如果为了便于管理，可以建立一个新的设计库用于lab2部分的实验，新建的库要attach到chrt35dg_COREcell工艺库上，以便获得相关的工艺信息
- chrt35dg_COREcell 库为IP core公司为Chart开发的标准单元库的版图，我们主要利用该库中的单元构建全加器。IOcells3L为IO单元库
- IOcells3L为IO单元库，现在不用

实验步骤2——确定逻辑单元

- 打开chrt35dg_COREcell库中的任意一个单元的版图，例如**AD01D1**，可以看到其版图数据，如果看不到正确的图层信息，请检查**display.drf**文件是否在**lab2**目录下
- 标准单元库中的**AD01D1**和**AD01D2**单元是库中的全加器，其中**D1**是一倍驱动能力，而**D2**是2倍驱动能力的单元，即驱动相同的负载电容，**D2**单元的速度较快，但是面积和功耗较多，通过版图可以看到二者面积的区别，逻辑综合器用高驱动能力的单元优化关键路径，而非关键路径使用低驱动能力单元以降低面积和功耗
- lab2**的实验目的是利用门级单元构建全加器，因此可以利用**除了**这两个已有的全加器以外的所有单元构建设计
- 其他的单元，大家可以参考单元库**说明文档**了解其逻辑功能和端口信号
- 根据全加器的真值表和逻辑表达式选择库中适当的单元实现其逻辑功能，这些单元的选择在**很大程度**上决定了后续版图设计，单元少则连线少，面积可能也小些

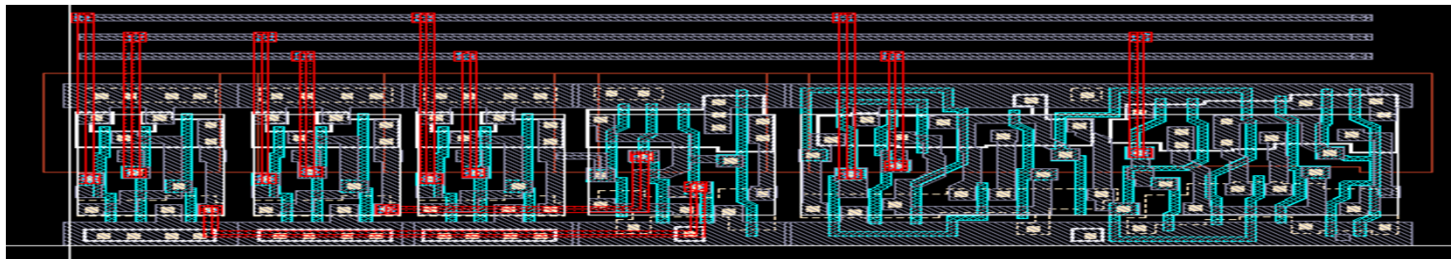
实验步骤3——版图设计：布局

- 由于标准单元库中没有提供电路原理图，我们直接利用其版图构建全加器，然后通过电路仿真验证其性能和功能
- 在设计库中建立一个新的layout cellview，例如叫fulladd，启动virtuoso layout editor工具，开始版图设计
- virtuoso LE工具是virtuosoXL之前的版本，为基于图形的定制版图工具，而virtuosoXL为基于连接关系的工具，由于门级设计中没有原理图，我们实验利用VLE进行设计
- 根据全加器的逻辑表达式，选择适当的库单元实现相关的逻辑功能，选择的原则是使得实现的版图面积最小，这一方面取决与单元的数目，另一方面取决与他们直接的连线
- 在VLE中例化组成全加器的所有单元，将其排成一行，最后放在横坐标上，单元直接邻接，可以共享vdd和gnd走线



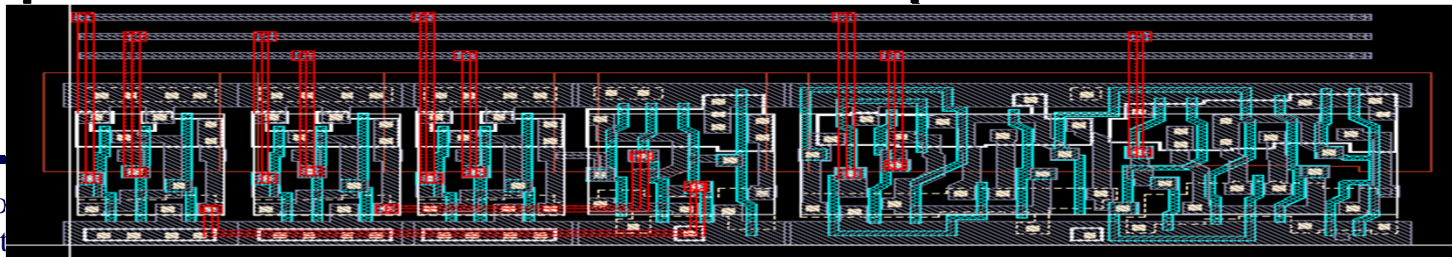
实验步骤3——版图设计：布线

- 对于完成布局后的版图设计，可以根据其逻辑连接关系进行连线，根据设计要求可以使用金属1和金属2进行走线
- 由于单元内部有很多金属1的走线，因此，一般原则是金属1在单元外部进行横向走线，金属2用于将单元内部的pin脚连接到单元外部的垂直走线，这样可以有效提高走线的效率
- 金属2也可以在单元之间的短距离的横向走线，当然在两个pin脚相邻较近的时候，金属1也可以在不短路和违反设计规则的前提下在单元内部横向走线
- 完成版图设计后进行drc检查，修正错误



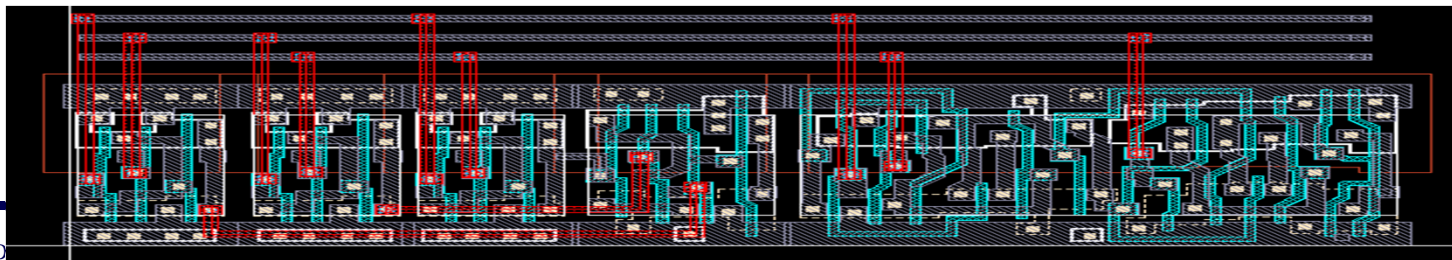
实验步骤3——版图设计：加pin脚

- 走线完成后，可以说版图设计已经完成，如果是**lab1**中正常的定制设计过程应该进行**lvs**检查；但是在**lab2**中，我们采用直接利用版图构建原型设计的方法，并没有进行原理图设计，也没有进行电路仿真，因此需要对完成后版图的功能和性能进行仿真验证，为此，首先需要提取版图的电路网表
- 版图上**pin**脚的目的是为了**让版图提取工具可以识别io信号的位置**，在完成后的版图上加**pin**脚，为后续的器件提取做好准备
- 走线完成后，在金属上加上**pin**，利用**create-pin**命令，加上全加器的输入和输出**pin**，以及电源**vdd!**和**gnd!**，**pin**的名称可以采用同**pin**脚相同的层名，也可以采用对应的**text**层，例如**A1**的**pin**用**A1TEXT**层作为**pin**名
- 加完**pin**以后可以再做一次**drc**，确认加**pin**的过程没有引入**drc**错误



实验步骤4——版图提取

- 为了进行电路仿真，需要对版图进行器件的电路网表提取
- 利用diva—extract工具进行提取，同lab1中过程一样，会生成全加器的extracted view，在library manager中打开该view，观察提取出来的器件和pin脚，器件较小，需要放大后才能看到，看看pin脚是否都提取出来，以及输入输出属性是否正确，如果不正确修正版图后重新进行提取
- 为了能够顺利仿真，还需要进行analog—extracted view的提取，为此，首先需要进行lvs，lvs的目的纯粹是为了能够提取出analog—extracted view，因此比较的对象都是extracted view
- 完成lvs后，利用lvs界面中的build analog命令提取出analog—extracted view，利用library manager可以看到生成的该view

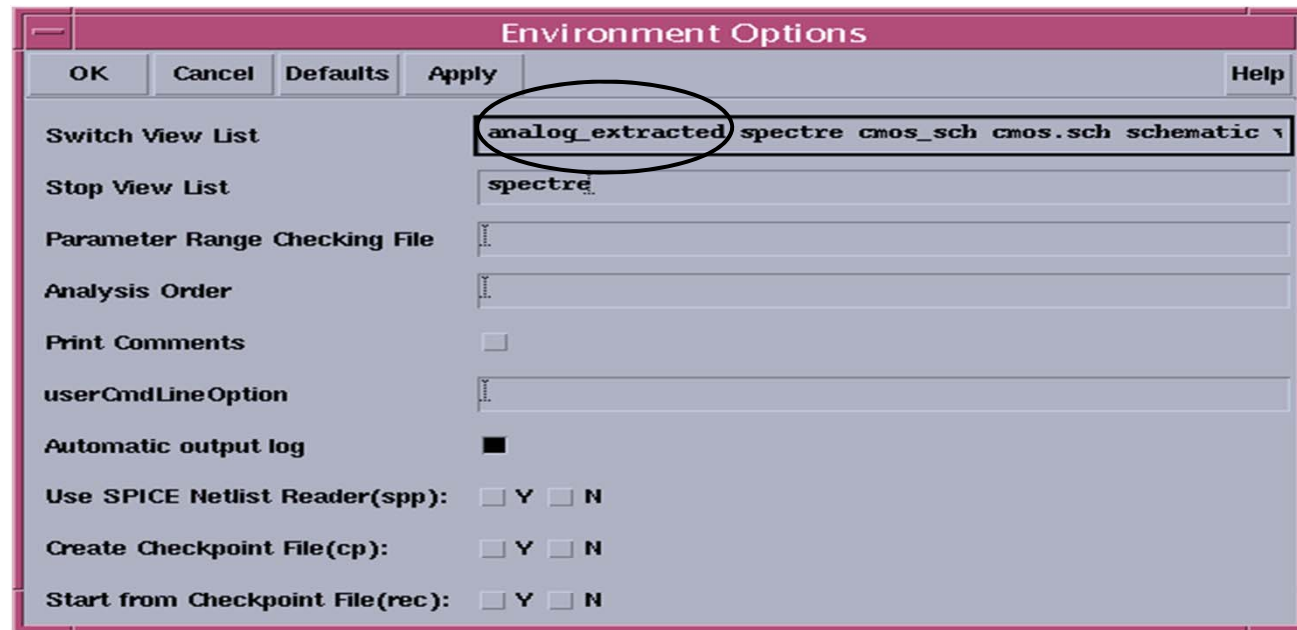


实验步骤5——电路仿真：仿真原理图

- 正常的定制设计过程应该是先做电路仿真和优化，然后进行版图设计和验证，lab2中基于现有的设计资源进行设计，电路仿真的目的主要是验证完成后版图的功能和性能
- 由于没有电路原理图，因此我们使用全加器版图提取后的analog—extracted view进行仿真
- 首先，建立一个伪的全加器原理图，只包括5个输入和输出pin脚，目的是为了生成symbol view，利用原理图生成symbol
- 然后，构建全加器的仿真电路图，例如叫testadd，例化全加器的symbol，添加激励源和电源，最后利用check and save命令（快捷键x）保存仿真电路图

实验步骤5——电路仿真：仿真配置

- 启动ADE，进行电路仿真，类似lab1中的过程，可以将lab1中存储的state载入，修改后进行仿真
- 仿真前，先打开Environment Options(setup->Environment)，在Switch View List 一栏的前面加入analog_extracted，以空格隔开
- 开始仿真



实验总结

- 由于是基于门电路的设计，无法进行器件级的优化，只能进行门级优化，即更好版图中的逻辑门
- 门级设计同lab1中的器件级设计相比，设计时间缩短，有利于快速构建原型设计
- 现代的布局布线工具可以帮助我们完成大规模电路的门级乃至更高层级的版图设计，我们在后面半定制的lab中会学习使用
- 如果有时间，大家可以进一步熟悉标准单元库中的单元，我们在后面的综合实验中可以选择合适的单元构建电路