

A world map is shown in the background, with vertical color bands of red, orange, yellow, green, blue, and purple overlaid on it. The map is dark, and the colors are vibrant. The text "www.docin.com" is overlaid on the map.

www.docin.com

数字IC设计流程与工具

Notes



- 本PPT内容是整个DDC项目组的集体学习研究成果
- 感谢已经毕业的曾经参与后端项目的师兄师姐，以及各位老师。
- 闻道有先后，术业有专攻
 - 共同学习，共同进步
 - 大家有问题请直接请教熟悉相应工具的同学。
- Tips: 可以参考QUATURS II的design flow!!

Contents



1

基于标准单元的ASIC设计流程

2

数字前端设计(front-end)

3

数字后端设计(back-end)

4

教研室ASIC后端文件归档

3

Q & A

Contents



- 1 基于标准单元的ASIC设计流程
- 2 数字前端设计(front-end)
- 3 数字后端设计(back-end)
- 4 教研室ASIC后端文件归档
- 3 Q & A

基于standcell的ASIC设计流程



数字前端设计。以生成可以布局布线的网表为终点。



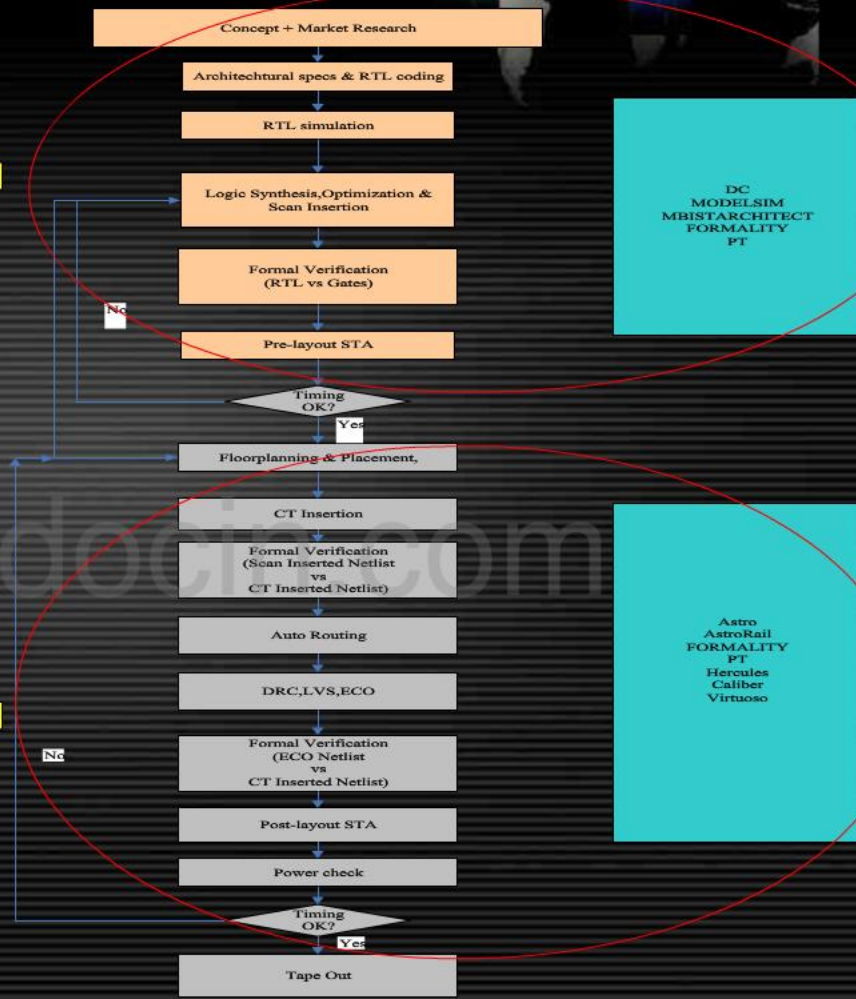
数字后端设计。以生成可以送交foundry进行流片的GDS2文件为终点。



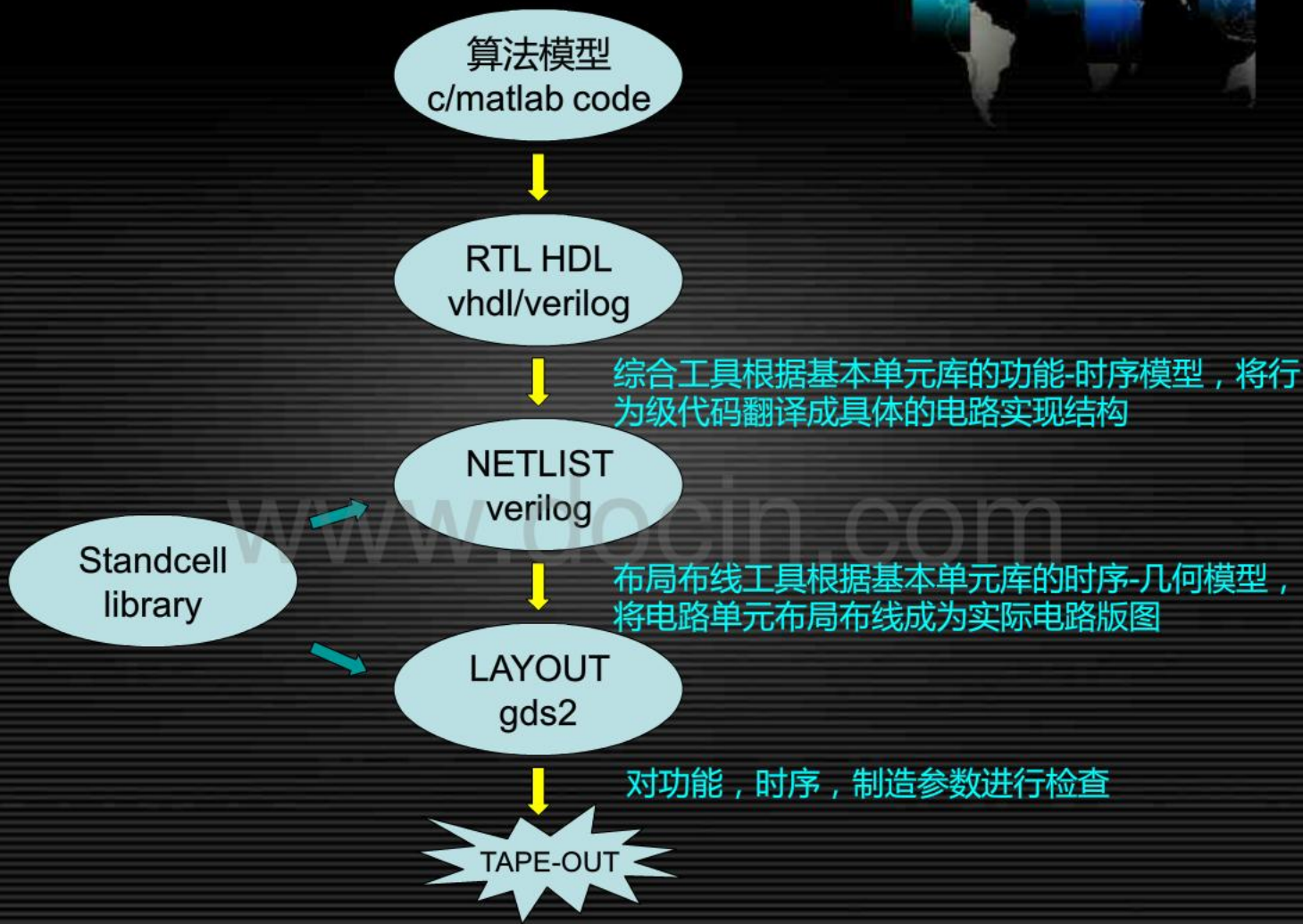
术语：

tape-out—提交最终GDS2文件做加工；

Foundry—芯片代工厂，如中芯国际。。。



基于standcell的ASIC设计流程



Contents



1

基于标准单元的ASIC设计流程

2

数字前端设计(front-end)

3

数字后端设计(back-end)

4

教研室ASIC后端文件归档

3

Q & A

数字前端设计流程-1



整个ASIC设计流程都是一个迭代的流程，在任何一步不能满足要求，都需要重复之前步骤，甚至重新设计RTL代码。

模拟电路设计的迭代次数甚至更多。。。

数字前端设计流程-2



! 怎样保证网表的正确性？

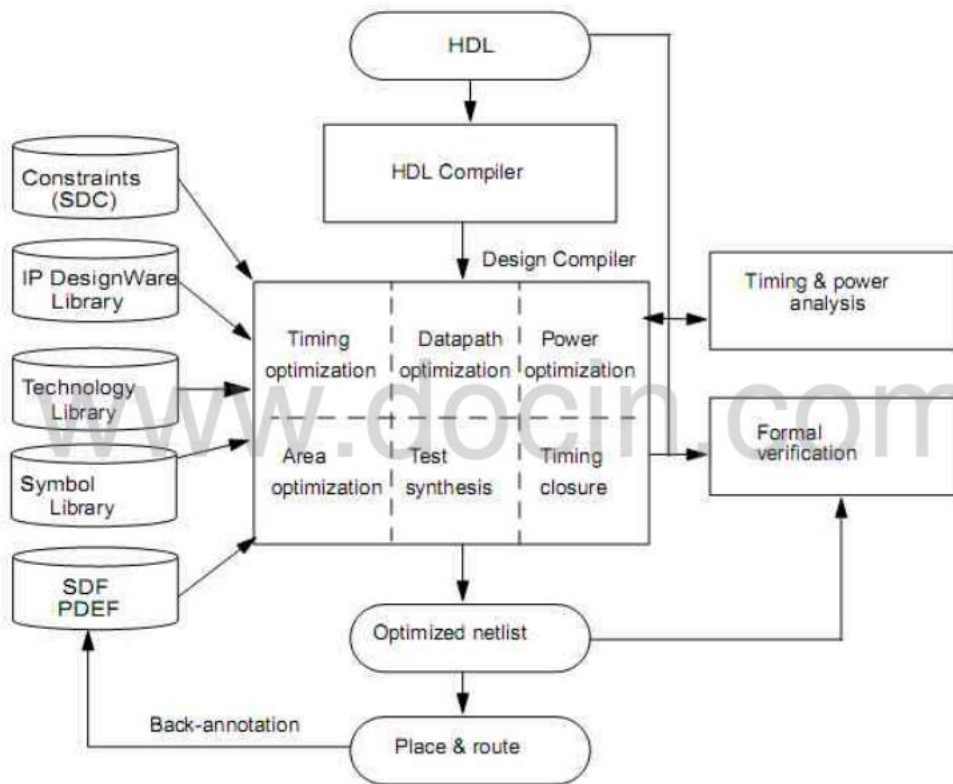
- 以往的方法是对网表文件做门级仿真。此种方式的仿真时间较长，且覆盖率相对较低。
- 形式验证+静态时序分析。此种方法仿真时间短，覆盖率高，为业界普遍采用的方式。

www.docin.com

数字前端设计流程-3 使用DC综合

●SYNOPSYS – Design Compiler

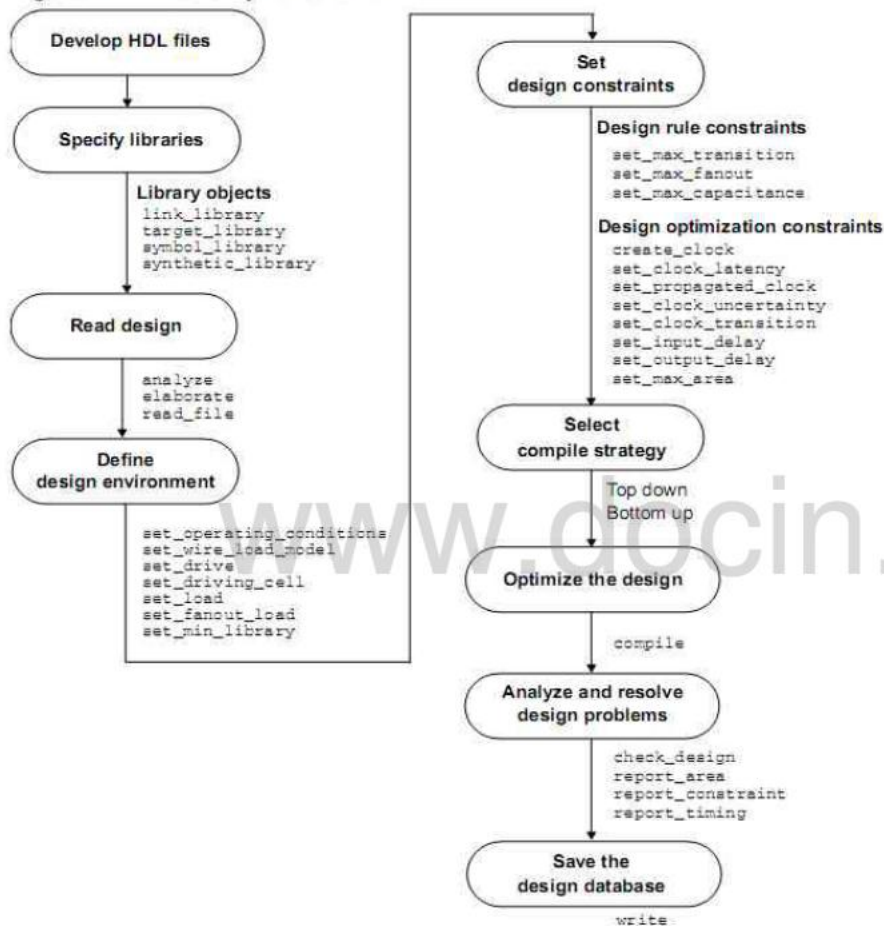
Figure 1-1 Design Compiler and the Design Flow



数字前端设计流程-4 使用DC综合



Figure 2-2 Basic Synthesis Flow



步骤可以归纳为：

- 1.指定综合使用的库
- 2.根据符号库将行为级模型转换为逻辑网表（由逻辑单元GTECH构成）
- 3.指定综合环境以及约束
- 4.进行综合，根据约束将逻辑网标映射为实际网表（由标准单元构成）
- 5.优化网表
- 6.输出综合结果

数字前端设计流程-5 使用DC综合



! ASIC的综合与FPGA的综合有什么不同？

- 原理是相同的！
- 关键在于综合目标不同。FPGA综合是将逻辑映射为FPGA器件资源（如LUT,REG,MEM-BLOCK）；ASIC综合是将逻辑映射为标准单元（如门电路，寄存器，RAM,ROM）。
- 标准单元库中对于某一种功能的门电路具有不同版本，分别对应不同驱动能力。

数字前端设计流程-6 使用DC综合



! 综合不仅仅要求功能，也要求时序！

- 综合具有一定条件，如工作频率、电路面积等。
- 门电路沟道宽度窄，自然面积小，但是驱动能力降低，电路工作速率降低。所以要对综合进行约束！
- 综合器中也有静态时序分析功能，用来计算当前综合结果的工作速率。
- 使用wire load model来估算延时。

数字前端设计流程-7 使用DC综合



! TIPS:

- 关于延时计算将在静态时序分析部分详细介绍。
- 可以参考QUATURS II软件的ANALYSIS & SYNTHESIS工具学习DC。

www.docin.com

数字前端设计流程-8 使用PT进行STA



- SYNOPSIS – Prime Time
- 只是一个时序分析工具，本身不对电路做任何修改。
- 在ASIC流程中对于电路进行任何修改过后都应该使用STA工具检查其时序，以保证电路时序满足要求。
- 仍然采用wire load model来估算电路时序。
- 可以参考QUATURS II的timequest timing analyzer学习。

数字前端设计流程-9 延时计算



- 采用wire load model可以计算电路端到端路径延时。
- 端到端路径：
 - 寄存器输出 >> 寄存器输入
 - 寄存器输出 >> 输出端口
 - 输入端口 >> 寄存器输出
- 延时采用标准单元库查表进行运算
 - Input : transition time, output net capacitance
 - Output : input to output delay , transition time
- Net capacitance 使用wire load model进行估算

数字前端设计流程-10 延时计算



```
lu_table_template(setup_template_3x3) {
    variable_1 : constrained_pin_transition;
    variable_2 : related_pin_transition;
    index_1 ("1000, 1001, 1002");
    index_2 ("1000, 1001, 1002");
}

lu_table_template(hold_template_3x3) {
    variable_1 : constrained_pin_transition;
    variable_2 : related_pin_transition;
    index_1 ("1000, 1001, 1002");
    index_2 ("1000, 1001, 1002");
}

lu_table_template(recovery_template_3x3) {
    variable_1 : constrained_pin_transition;
    variable_2 : related_pin_transition;
    index_1 ("1000, 1001, 1002");
    index_2 ("1000, 1001, 1002");
}

lu_table_template(removal_template_3x3) {
    variable_1 : constrained_pin_transition;
    variable_2 : related_pin_transition;
    index_1 ("1000, 1001, 1002");
    index_2 ("1000, 1001, 1002");
}
```

```
cell (DFFRHQX1) {
    cell_footprint : dffrhq;
    area : 35.645400;
    pin(D) {
        timing() {
            related_pin : "CK";
            timing_type : setup_rising;
            rise_constraint(setup_template_3x3) {
                index_1 ("0.042, 0.392, 1.5");
                index_2 ("0.042, 0.392, 0.764");
                values ( \
                    "0.164062, 0.101562, 0.089844", \
                    "0.218750, 0.156250, 0.144531", \
                    "0.242188, 0.179688, 0.164062");
            }
            fall_constraint(setup_template_3x3) {
                index_1 ("0.042, 0.392, 1.5");
                index_2 ("0.042, 0.392, 0.764");
                values ( \
                    "0.164062, 0.093750, 0.058594", \
                    "0.277344, 0.207031, 0.171875", \
                    "0.531250, 0.460937, 0.425781");
            }
        }
    }
}
```

数字前端设计流程-11 延时计算



- 布局布线前，由于无布线信息，所以连线延时只能够通过连接关系（与fanout相关）估计得到。
- 当特征尺寸降低时，此种估计方法越来越不准确，所以可以使用physical synthesis技术。
- 在布局布线后，布局布线工具可以提取出实际布线后的线网负载电容，此时PT可以计算实际延时(back-annotate)。
- 可以将延时信息写入SDF (synopsys delay file)文件用于后仿真。

数字前端设计流程-12 PT使用流程



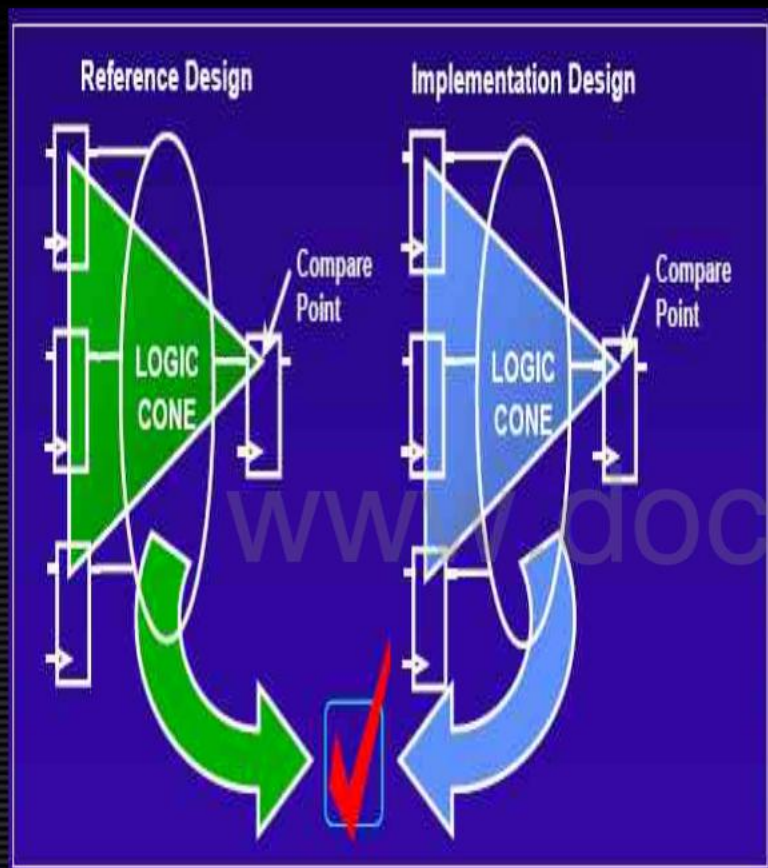
- 使用方法与DC类似
 - 1.指定使用的库
 - 2.读入网表文件
 - 3.指定时序约束及工作环境
 - 4.进行静态时序分析，给出报告
- 从一个synthesizable subcircuit 中，pt能捕获一个时序环境，并写成一系列的dc指令，在dc中用其为此subcircuit定义时间约束和时序优化 - - 值得关注
- 这两个都支持用SDC (synopsys design constraints) 格式指定设计规则，包括时间面积约束。

数字前端设计流程-13 形式验证



- 静态时序分析检查了电路时序是否满足要求，而形式验证检查了电路功能的正确性。
- 形式验证工具本质是一个比较器！其功能就是比较两电路功能是否完全一致。
- 由于在综合过程中电路节点名称可能改变，因此可以使用形式验证工具找到RTL代码中节点在网表中的对应节点。

数字前端设计流程-14 逻辑锥



●原理

把设计划分成无数个逻辑锥 (logic cone) 的形式, 以逻辑锥为基本单元进行验证. 当所有的逻辑锥都功能相等, 则验证 successful !

●逻辑锥

锥顶作为比较点. 它可以由原始输出, 寄存器输入, 黑盒输入充当
---- formality自动划分

数字前端设计流程-15 形式验证



! 什么时候需要做形式验证？

- Verify RTL designs vs. RTL designs
 - the rtl revision is made frequently
- Verify RTL designs vs. Gate level netlists
 - verify synthesis results
 - verify manually coded netlists, such as Design Ware
- verify Gate level netlists vs. Gate level netlists
 - test insertion
 - layout optimization

Contents



1

基于标准单元的ASIC设计流程

2

数字前端设计(front-end)

3

数字后端设计(back-end)

4

教研室ASIC后端文件归档

3

Q & A

数字后端设计流程-1



目前业界广泛使用的APR(Auto Place And Route)工具有：

- Synopsys公司的ASTRO
- Cadence公司的Encounter

www.docin.com

可以参考QUARTUS II的FITTER学习。

数字后端设计流程-2



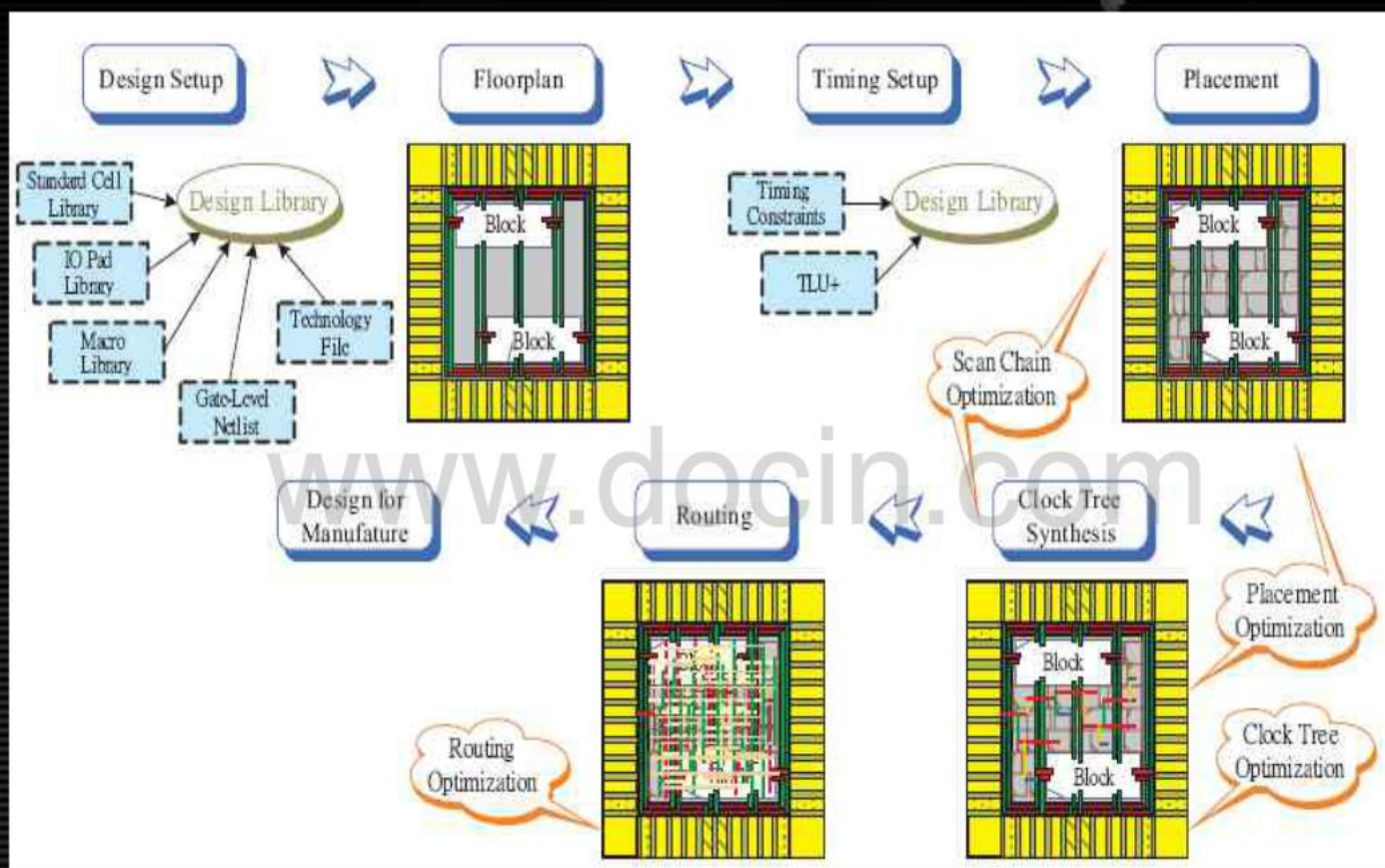
! 哪些工作要APR工具完成？

- 芯片布图（RAM,ROM等的摆放、芯片供电网络配置、I/O PAD摆放）
- 标准单元的布局
- 时钟树和复位树综合
- 布线
- DRC
- LVS
- DFM（Design For Manufacturing）

www.docin.com

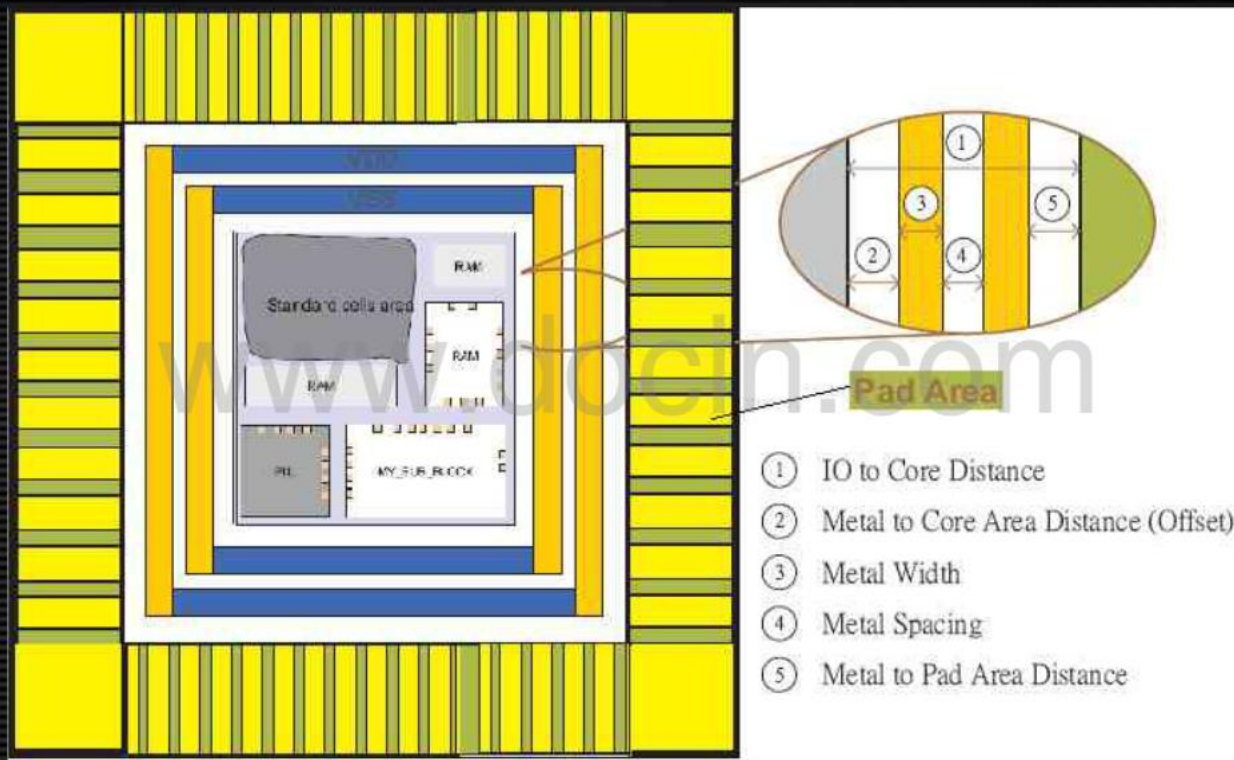
数字后端设计流程-3

ASTRO布局布线流程

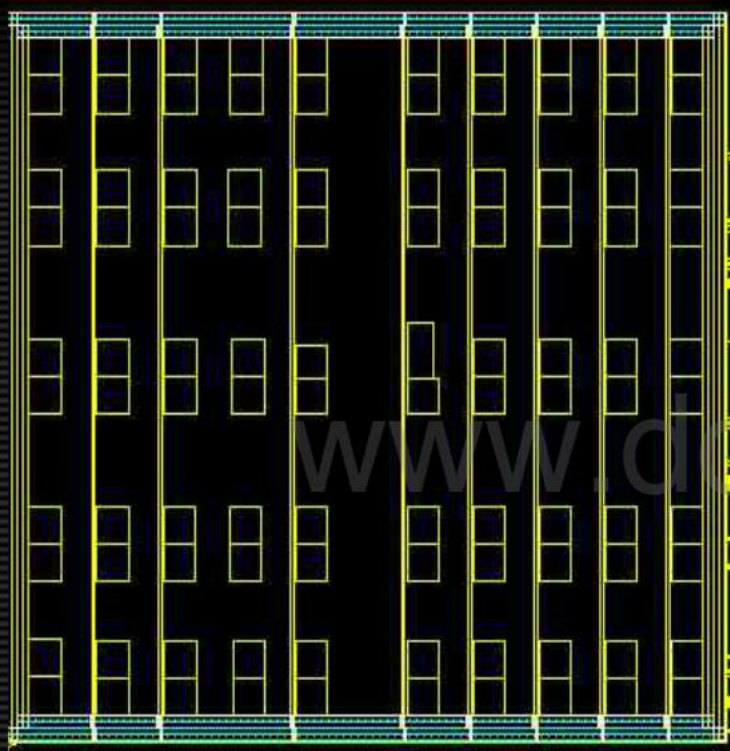


数字后端设计流程-4 布图

布图步骤主要完成宏单元的放置，电源规划以及PAD的摆放，布图影响到整个设计的绕线难易以及时序收敛。



数字后端设计流程-4 布图



电源环的宽度计算：

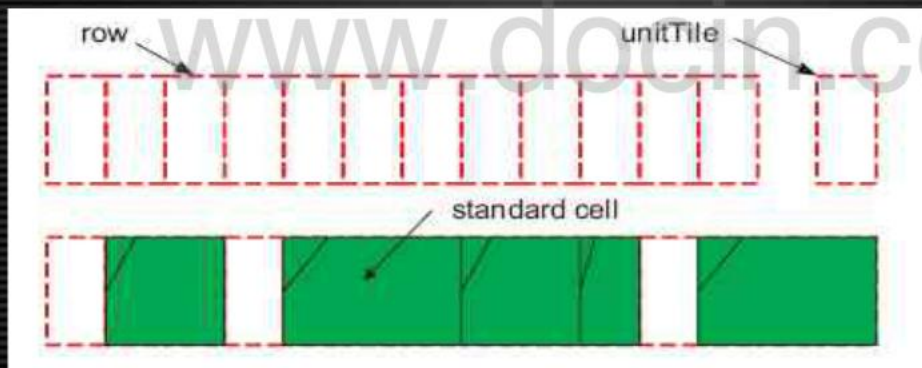
$$W = \frac{I_p}{I_{\max} \times 4 \times n} \times k$$

www.docin.com

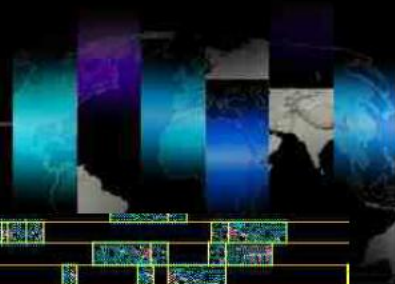
数字后端设计流程-5 布局



- Astro是一个grid based软件，grid 分为 placement grid和routing grid.
- Placement grid就是所谓的unitTile, unitTile 为一个row的最小单位，standard cell 就是摆放在row上面，起摆放位置须对齐每个unitTile的边缘，因此每个standardcell都必须是同一高度。



数字后端设计流程-5 布局

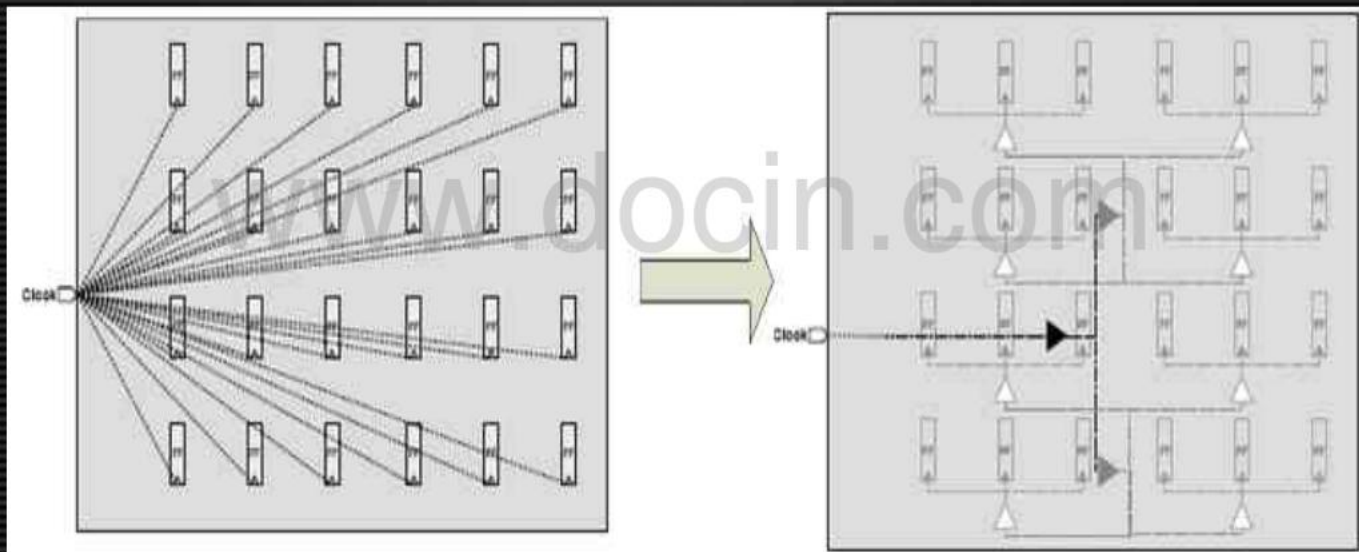


www.hic.in.com

数字后端设计流程-5 时钟树和复位树综合

时钟树综合的目的：

- 低skew
- 低clock latency



数字后端设计流程-6 时钟树和复位树综合



! 时钟树和复位树综合为什么要放在APR时再做呢？

- 在DC综合时并不知道各个时序元件的布局信息，时钟线长度不确定。
- DC综合时用到的线载模型并不准确。

www.docin.com

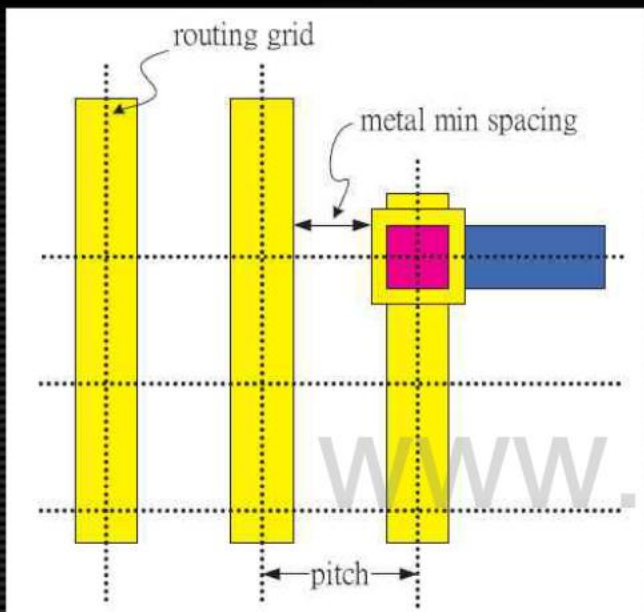
数字后端设计流程-7 布线



将分布在芯片核内的模块、标准单元和输入输出接口单元（I/O pad）按逻辑关系进行互连，其要求是百分之百地完成他们之间的所有逻辑信号的互连，并为满足各种约束条件进行优化。

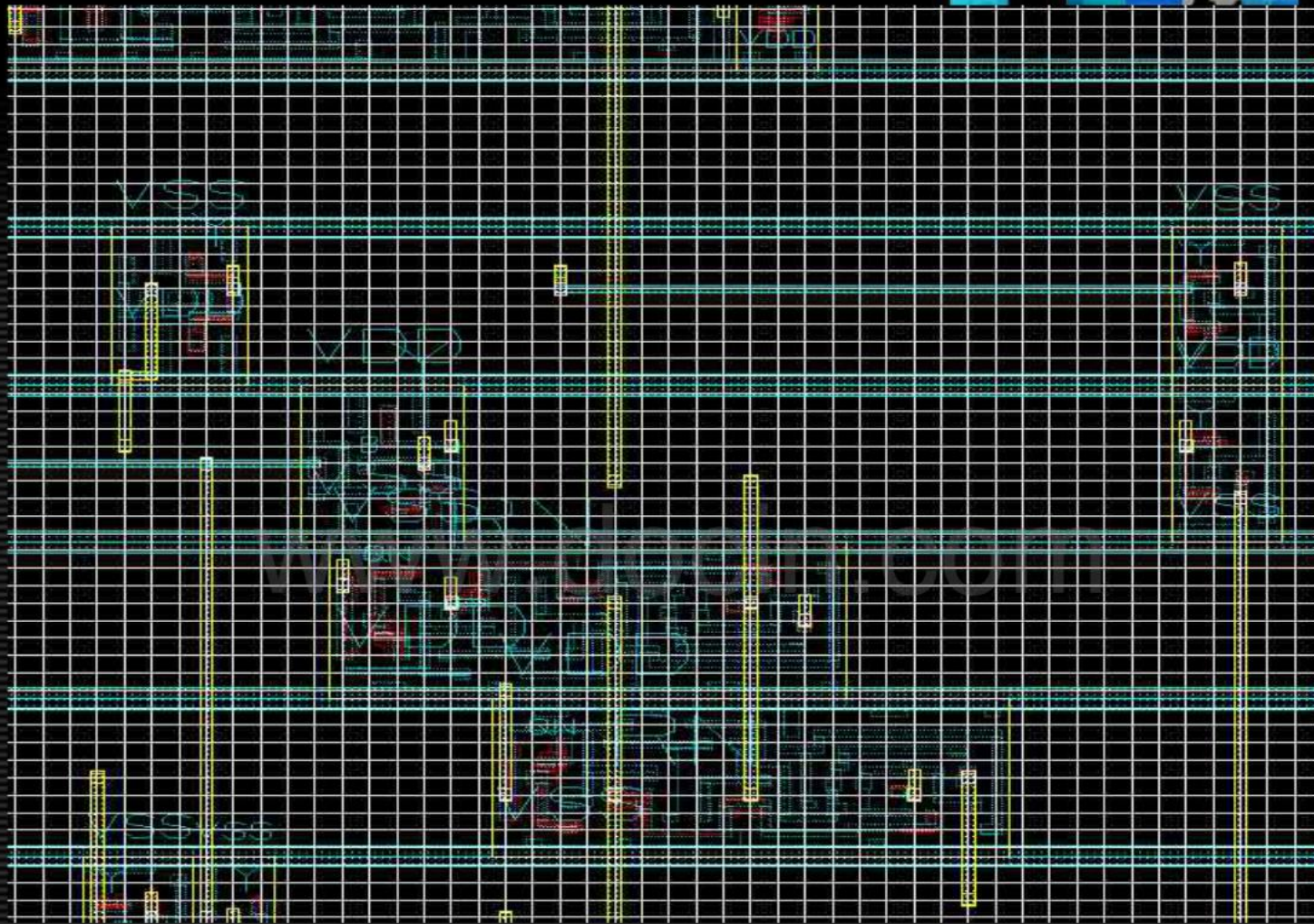
布线工具会自动进行布线拥塞消除、优化时序、减小耦合效应、消除串扰、降低功耗、保证信号完整性等问题。

数字后端设计流程-8 布线



Layer	"METAL1"	pitch	= 0.41
Layer	"METAL2"	pitch	= 0.46
Layer	"METAL3"	pitch	= 0.41
Layer	"METAL4"	pitch	= 0.46
Layer	"METAL5"	pitch	= 0.41
Layer	"METAL5"	pitch	= 0.46
Layer	"METAL7"	pitch	= 0.41
Layer	"METAL8"	pitch	= 0.96

数字后端设计流程-8 布线

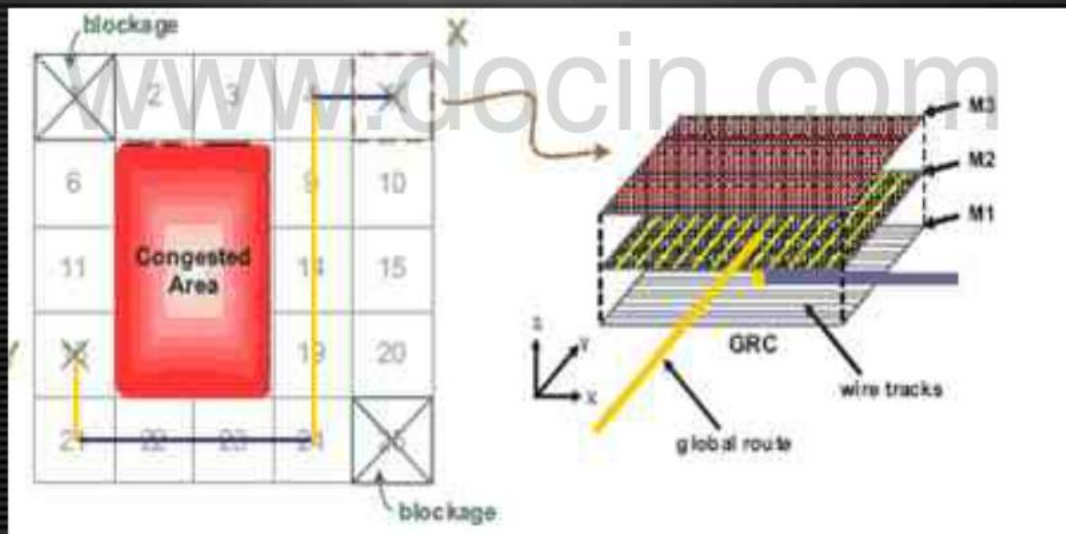


数字后端设计流程-8 布线



第一步 全局布线

Global route 进行时，整个芯片会被切割成一块块的global routing cell (GRC)，其目的在于建立一个绕线的蓝图。对于每个GRC，Astro会去计算包含其中且可以使用的wire track，根据这些信息选择绕线要经过的GRC。如图所示，有一个以X为起点Y为终点的连接需要绕线，考虑到blockage和congestion的状况后，选择了变化4、9、14、19、24、23、22、21、16的GRC来绕线。



数字后端设计流程-9 布线



第二步 布线通道分配

在global route 时已经将信号线分配到每个GRC，而track assignment的功能就是将这些信号线在分配到每个track上，决定每条线要走的路径。Track assignment是以整个芯片为处理单位来作规划，尽量绕出又长又直且via数目最少的绕线。

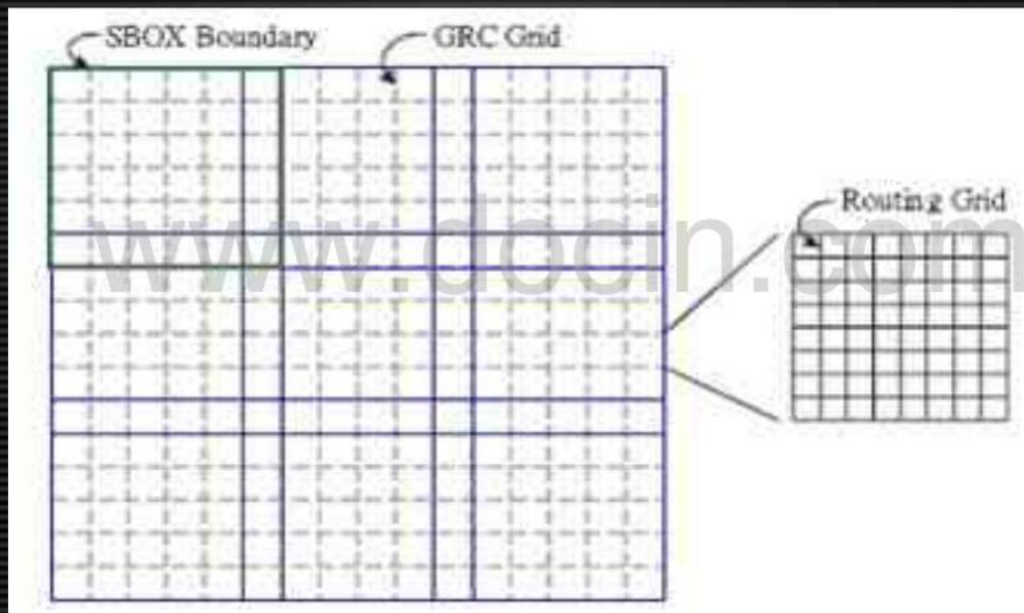
www.docin.com

数字后端设计流程-10 布线



第三步 详细布线

Detail route的工作主要是将track assignment的DRC violation移除，一次是以一个switch box (SBOX)为单位来进行修复的。SBOX由GRC构成，且每个SBOX的边缘会重叠一个GRC的宽度。



数字后端设计流程-11 DFM



! 什么是DFM呢？

- DFM : Design For Manufacturing
- DFM步骤在整个布局布线流程以后开始，主要目的是通过一些技术处理防止芯片在物理制造过程中出现问题，造成芯片不能工作。DFM的目的在于提高良率。

DFM包括：

- 天线效应（信号线太长造成）
- Metal liftoff效应防止（由金属密度过大造成）
- Metal over-etching效应防止（由金属密度过低造成）

数字后端设计流程-12



基于标准单元的APR布局布线与FPGA有什么区别？

- 基本原理是一样的
- FPGA内部的逻辑单元以及走线资源都是固定的，布局布线工具只是完成如何使用这些资源以使得整个设计收敛。
- 而基于标准单元的APR时，标准单元位置以及走线资源都是可以根据需要调整的，因此灵活性更大，更容易使得整个设计收敛。

数字后端设计流程-13 DRC



- DRC – Design Rule Check

- 何谓Design Rule

- 由于制造工艺与电路性能等原因，对版图设计有一定要求，比如说，线宽不能低于最低线宽，N阱间应当具有一定间距，每一层金属应当具有一定密度等。等等。

- 天线规则：当版图中的金属线具有一定长度时，会造成天线效应。因此需要对自动APR工具的布线做检查。

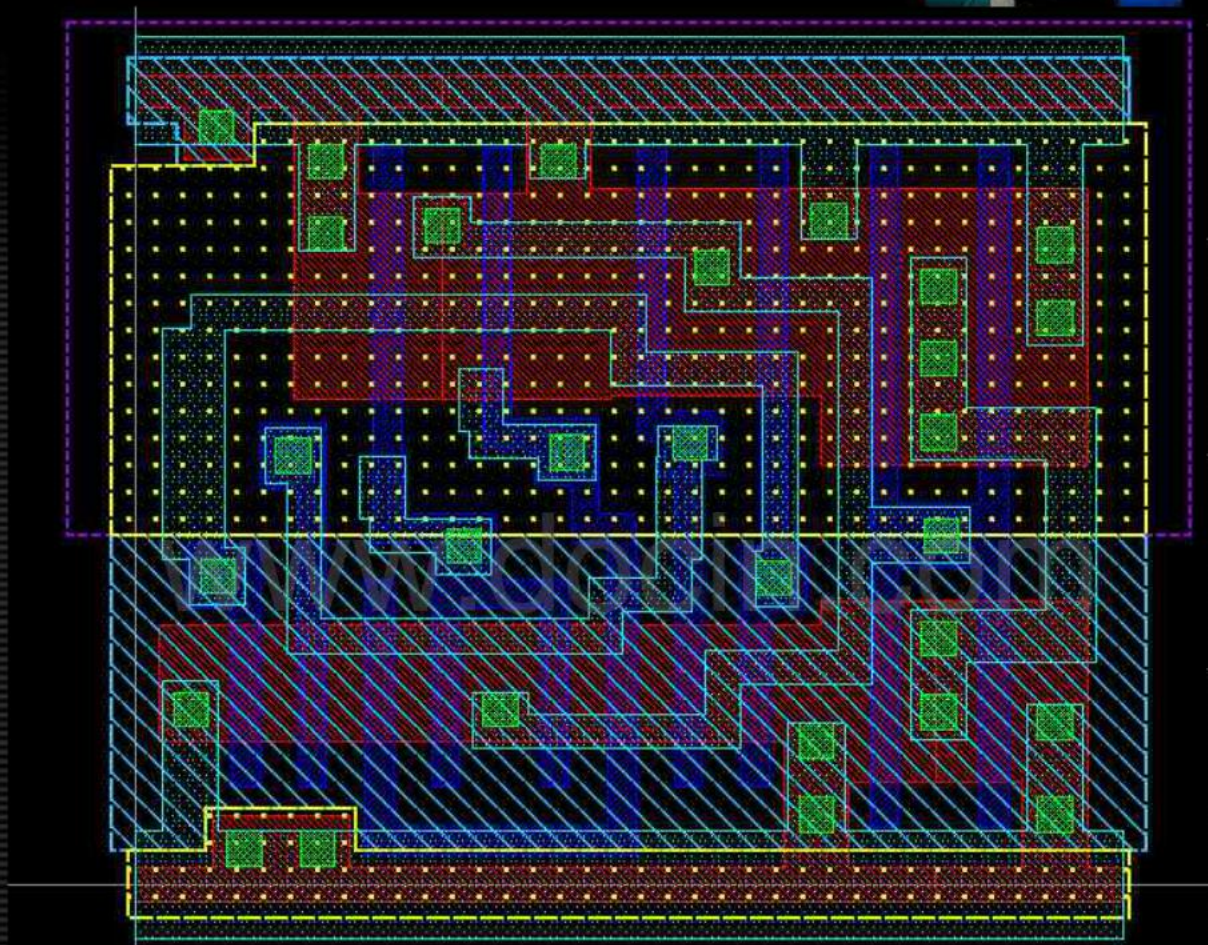
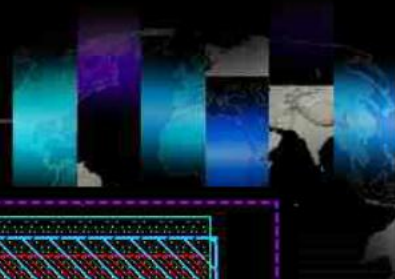
- DUMMY：由于制造工艺要求每一层金属必须具有一定密度，因此需要工具自动往空余部分填充冗余金属。

数字后端设计流程-14 DRC



- DRC原理：基于计算机图形学！
- 版图中的不同结构可以表示为不同的层，如：
 - N阱
 - P阱
 - 栅
 - 各层金属线
- 版图中的每一个电路原件与连接线均由一系列具有一定大小，位于相应位置的矩形构成。
- 规则检查则建模为图形性质计算

数字后端设计流程-14 DRC



数字后端设计流程-15 LVS



- LVS – layout vs schematic
- LVS是为了检查版图文件功能与原有电路设计功能的一致性。
- LVS的原理：网表比对！
 - 参考网表为APR工具时钟树、复位树综合后的网表。-- HDL文件
 - 比对网表为LVS工具从版图中提取电路元件以及连接关系以后得到的网表
- LVS软件根据标准单元库设计者提供的cdl网表文件从版图中提取电路网表。

数字后端设计流程-16 LVS



! 什么时候需要做DRC/LVS ?

只要对版图信息做修改，就需要做DRC/LVS检查。

www.docin.com

数字后端设计流程-17 CALIBRE



- MENTOR GRAPHIC – CALIBRE
 - 专业的DRC/LVS软件，可以单独使用，也可以嵌入virtuoso,astro中联合使用。
 - 使用foundry提供的DRC/LVS检查脚本，可以自动完成DRC/LVS工作，且给出错误报告。
 - 检查出的错误需要在版图编辑工具中修改。
- www.docin.com
- DRC/LVS工具还有DIVA,DRACURA等。。。

数字后端设计流程-18 VIRTUOSO



- CADENCE – VIRTUOSO
- 专业版图编辑工具，结合CALIBRE可以对版图做在线检查修改。
- 使用方式与PROTEL类似。。
- VIRTUOSO生成最终流片版图

www.docin.com

数字后端设计流程-19 SIGN-OUT



- 当设计完成时，应当保证其时序，功能，工艺等指标完全达到要求，只要有任意一点不能达标，便需要重新对设计做修改！
- Foundry为了规避责任，故要求设计者在提交版图时签字画押，说明此版图已经经过检查，是没有错误的。。。www.docin.com
- 流片很贵，大家应慎之又慎。。。www.docin.com

Contents



1

基于标准单元的ASIC设计流程

2

数字前端设计(front-end)

3

数字后端设计(back-end)

4

教研室ASIC后端文件归档

3

Q & A

教研室现有的ASIC工具软件



- SYNOPSYS

- Design Compiler 2004, Design Compiler 2006

- Prime Time 2004, Prime Time 2006

- Formality 2004, Formality 2006

- Astro 2004, Astro 2006

- CADENCE

- Virtuoso ic5451

- MENTOR GRAPHIC

- Calibre 2005



以及相应破解和教程

教研室现有的基本单元库



- SMIC 0.13um元件库，IO PAD 库，包括
 - 综合，STA所用的.lib行为模型文件
 - 布局布线所用元件版图文件
 - 技术库（.tf,virtuoso使用）
 - 模型网表文件(.cdl)
 - 相应说明pdf
- ARTISAN RAM/ROM compiler

Q & A



- 数字前端：钱宇平，郑昕，郑宇，杨一波
- 数字后端：赵文豪，潘经纬
- 还可以问：刘科，刘欣，王磊，陈星宇，詹璨铭。。。

www.docin.com

