

Technology Files

EE291F

Speaker Yuxuan Li (liyx22025@shanghaitech.edu.cn)

VSPLab@SIST



上海科技大学

ShanghaiTech University

Contents

目录

1 工艺库文件导入

2 工艺库文件分析

工艺库文件导入

工艺库文件导入

工艺库（Technology Library）是指某个半导体工艺所配套的一整套设计数据。

- 本课程中，使用的是 HL55 的工艺库，库内容根据课程需求进行了裁切和重排。
- 本课程示例的工艺库目录位于 `/mnt/nfs_share/EE291F/3003/tech.lab`。
- **严禁出于非课程学习目的下载并传播该工艺库，一经发现，严惩不贷！**

立志成才 报国裕民

工艺库文件导入

推荐将工艺库目录软链接到项目目录中，命名为 tech，便于后续使用和访问。

在 Makefile 中写入以下内容

```
TECH_SRC:=/mnt/nfs_share/EE291F/3003/tech.lab
TECH_DIR:=tech
.PHONY: init
init:
    ln -sfn ${TECH_SRC} ${TECH_DIR}
```

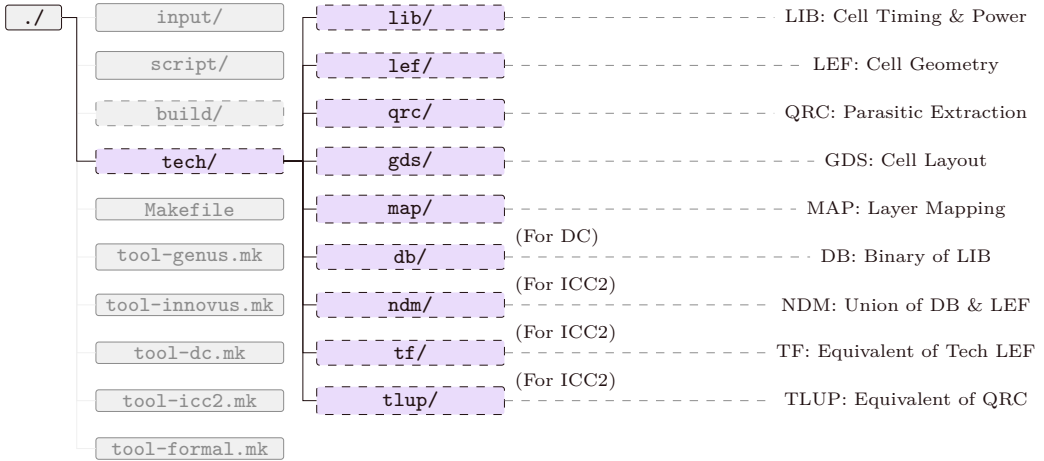
随后运行

```
make init
```

其中，-s 代表软链接，-f 代表允许覆盖，-n 代表允许覆盖目录软链。

工艺库文件分析

工艺库文件总览


./
input/
script/
build/
tech/
Makefile
tool-genus.mk
tool-innovus.mk
tool-dc.mk
tool-icc2.mk
tool-formal.mk
lib/
lef/
qrc/
gds/
map/
db/
ndm/
tf/
tlup/
LIB: Cell Timing & Power
LEF: Cell Geometry
QRC: Parasitic Extraction
GDS: Cell Layout
MAP: Layer Mapping
(For DC)
DB: Binary of LIB
(For ICC2)
NDM: Union of DB & LEF
(For ICC2)
TF: Equivalent of Tech LEF
(For ICC2)
TLUP: Equivalent of QRC

工艺库文件总览

在这一部分，我们主要研究两个问题

- `tech` 目录下的每个目录代表一类工艺库文件，它们分别是做什么的?
- `tech.tcl` 应该如何妥当组织，使后续各工具的 Tcl 可以便捷地访问库文件?

立志成才 报国裕民

LIB：描述单元的时序与功耗信息。

tech/lib


tech/
lib/
lef/
qrc/
gds/
map/
db/
ndm/
tf/
tlup/
std/
mem/
M31GPSC900HL055PR_N40CFF1P32_cbest_ccs.lib
M31GPSC900HL055PR_125CSS1P08_cworst_ccs.lib
M31GPSC900HL055PL_N40CFF1P32_cbest_ccs.lib
M31GPSC900HL055PL_125CSS1P08_cworst_ccs.lib
mem55lpw1024d16sp_ff1p32vn40c.lib
mem55lpw1024d16sp_ss1p08v125c.lib
mem55lpw256d16sp_ff1p32vn40c.lib
mem55lpw256d16sp_ss1p08v125c.lib
LIB: STD, RVT, FF/−40 °C/1.32 V
LIB: STD, RVT, SS / 125 °C/1.08 V
LIB: STD, LVT, FF/−40 °C/1.32 V
LIB: STD, LVT, SS / 125 °C/1.08 V
LIB: MEM, 1024, FF/−40 °C/1.32 V
LIB: MEM, 1024, SS / 125 °C/1.08 V
LIB: MEM, 256, FF/−40 °C/1.32 V
LIB: MEM, 256, SS / 125 °C/1.08 V

两类单元

- **标准单元**（Standard Cell）：包含逻辑门和触发器。具有统一的几何规格，高度一致，宽度可变，在版图上像积木一样按行摆放，由工具自动布局布线，是数字电路实现的基本单元。
- **存储器宏**（Memory Macro）：存储器需要大规模阵列化的元件和电路，且位宽和深度往往是根据设计而变的。实践中的做法是，工艺库提供不同类型存储器的模板文件，设计者会通过专用的 Memory Compiler 工具（例如 Synopsys 的 Integrator）编译出特定参数的存储器，并在 RTL 中直接实例化这类存储器宏。**课程的工艺库提供的是预编译好的存储器**。
- 课程示例使用了两个存储器宏：1024×16 bit 和 256×16 bit

立志成才 报国裕民

标准单元的阈值和尺寸

标准单元可以通过**阈值**和**尺寸**换取不同的对速度的折中优化

表: 阈值和尺寸

项	阈值	尺寸
表现形式	HVT、RVT、LVT	X1、X2、X4
影响方式	阈值影响开关电压	尺寸影响驱动电流
影响结果	阈值越低，逻辑门越快	尺寸越大，逻辑门越快
折中代价	阈值越低，静态功耗越大	尺寸越大，动态功耗和面积越大
工艺库粒度	不同阈值对应不同库文件	不同尺寸位于同一库文件

该课程的工艺库仅提供了常规阈值（RVT）和低阈值（LVT）的标准单元。

立志成才 报国裕民

时序库的 PVT 角：工艺、电压、温度

由于芯片制造工艺的误差和芯片使用环境（不同的电压，不同的温度）的变化，需确保设计在不同的**PVT 角——工艺**（Process）、**电压**（Voltage）、**温度**（Temperature）下都可以正常工作！

因此，工艺库需要提供一个单元在不同 PVT 角下的时序信息（反映为多个 LIB 文件）。

表: 工艺、电压、温度

项	工艺（NMOS/PMOS）	电压	温度
最快	快角（FF）	高压（1.32 V）	低温（$-40\,^\circ$C）
最慢	慢角（SS）	低压（1.08 V）	高温（ 125 °C）
常规	常规角（TT）	常压（1.20 V）	常温（ 25 °C）

简单总结：**最快来自于 FF 和高压低温的组合**、**最慢来自于 SS 和低压高温的组合**。

该课程的工艺库仅提供了最快（FF 高压低温）和最慢（SS 低压高温）的 PVT 角。

立志成才 报国裕民

时序库的 PVT 角：工艺、电压、温度

然而，温度对快慢的影响实际上有两种相反的机制。在 HL55 的库上，低温快，高温慢，是正确结论。**但普遍情况是，不同器件，不同型号，不同电压，最快和最慢的温度都是不同的**！在完整的工艺库中，往往会为“FF 高压”和“SS 低压”的组合分别再提供各个温度的时序库。

课程只提供两个 PVT 角，是出于课程简化和 HL55 特殊的温度简单性下的选择。

立志成才 报国裕民

库格式：CCS 和 NLDM

如果你真的查看了 `tech/lib/std`，你可能会注意到其实有 `ccs` 和 `nldm` 两组库文件[1]

- CCS 和 NLDM 是两种不同的时序描述模型，它们都可以包装在 LIB 格式下。
- 精度低，但体积小速度快：**非线性延时模型**（Non Linear Delay Model, **NLDM**）
- 精度高，但体积大速度慢：**复合电流源模型**（Composite Current Source, **CCS**）

该课程统一使用 CCS 模型的库。不过，课程工艺库仍然同时提供了两种格式的标准单元库。部分原因是，单个 NLDM 库的体积大约是 $65\,\mathrm{MB}$，单个 CCS 库的体积大约是 $951\,\mathrm{MB}$，作为文本文件，前者尚且勉强能用常规的文本编辑器打开，后者如果是用 VSCode 打开，会直接崩溃！

[1] Memory 的库文件名称没有标注模型，但其实际也是 CCS 模型。

立志成才 报国裕民

我们当然可以直接列出每一个库文件的路径。但更为明智的做法是，观察库文件名称中的规律性，将其拆分为若干个字段，定义为变量，随后再通过这些变量组合出完整的库文件路径。[2]

```
set N_STD(RVT)      M31GPSC900HL055PR
set N_STD(LVT)      M31GPSC900HL055PL
set N_STD(FF)       N40CFF1P32_cbest_ccs
set N_STD(SS)       125CSS1P08_cworst_ccs
```

```
set N_MEM(M1024)    mem55lpw1024d16sp
set N_MEM(M0256)    mem55lpw256d16sp
set N_MEM(FF)       ff1p32vn40c
set N_MEM(SS)       ss1p08v125c
```

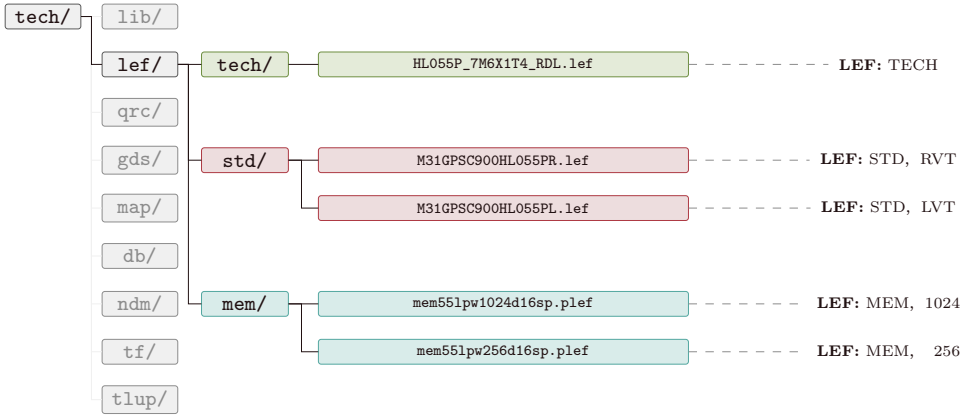
[2] `N_STD(RVT)` 和 `N_STD(LVT)` 是数组语法，此处这样用只是想避免 `N_STD_RVT` 和 `N_STD_LVT` 语义上不清晰。

我们会将同类型的库文件组成列表，便于使用。这里 [concat ...] 用于连接多个列表。

```
set LIB_STD_RVT_FF    [file join ${LIB_DIR} std ${N_STD(RVT)}_${N_STD(FF)}.lib]
set LIB_STD_RVT_SS    [file join ${LIB_DIR} std ${N_STD(RVT)}_${N_STD(SS)}.lib]
set LIB_STD_LVT_FF    [file join ${LIB_DIR} std ${N_STD(LVT)}_${N_STD(FF)}.lib]
set LIB_STD_LVT_SS    [file join ${LIB_DIR} std ${N_STD(LVT)}_${N_STD(SS)}.lib]
set LIB_MEM_M1024_FF [file join ${LIB_DIR} mem ${N_MEM(M1024)}_${N_MEM(FF)}.lib]
set LIB_MEM_M1024_SS [file join ${LIB_DIR} mem ${N_MEM(M1024)}_${N_MEM(SS)}.lib]
set LIB_MEM_M0256_FF [file join ${LIB_DIR} mem ${N_MEM(M0256)}_${N_MEM(FF)}.lib]
set LIB_MEM_M0256_SS [file join ${LIB_DIR} mem ${N_MEM(M0256)}_${N_MEM(SS)}.lib]
set LIB_LIST_FF       [list ${LIB_STD_RVT_FF} ${LIB_STD_LVT_FF} \
                            ${LIB_MEM_M1024_FF} ${LIB_MEM_M0256_FF}]
set LIB_LIST_SS       [list ${LIB_STD_RVT_SS} ${LIB_STD_LVT_SS} \
                            ${LIB_MEM_M1024_SS} ${LIB_MEM_M0256_SS}]
set LIB_LIST          [concat ${LIB_LIST_FF} ${LIB_LIST_SS}]
```

LEF：描述单元的几何信息。

tech/lef



立志成才 报国裕民

两种 LEF 文件

LEF 文件实际上可以分为两种：**单元 LEF**（Cell LEF）、**工艺 LEF**（Tech LEF）。

表: 单元 LEF 和工艺 LEF

项	单元 LEF	工艺 LEF
描述内容	单元的几何外形：外框、引脚、阻挡	工艺的金属定义：层定义、层规则、单位
工艺库粒度	每类单元一个库	每种金属栈一个库

立志成才 报国裕民

金属栈

金属栈（Metal Stack）定义了用于走线的金属层的数量和类型。就像 PCB 可以有双层板和四层板，工艺库往往会提供多个不同的金属栈选择，复杂的设计会需要更多的金属层才能走通。

课程工艺库只提供了一个金属栈的工艺 LEF，观察名称 `HL055P_7M6X1T4_RDL.lef`

- 该命名的一般规律是 `nM xX tTk`
- `nM` 指金属层的总数量（`7M`：7 层金属）。
- `xX` 指常规金属层的数量（`6X`：6 层常规金属层 `M1`，`M2`，`M3`，`M4`，`M5`，`M6`）
- `tT` 指厚金属层的数量，`k` 指其最小线宽倍数（`1T4`：1 层四倍线宽厚金属层 `T4M2`）[3]

在布线时，我们需要根据布线的类型（电源/时钟/逻辑）向工具指定合适的层范围。

[3]有趣的是，尽管这里只有一个厚金属层，但它的名称是 `T4M2` 而不是 `T4M1`。

金属栈

金属层只能横向或纵向走线！这是 ASIC 和 PCB 间一个很大的差异

- 每个金属层有自己的首选方向。沿首选方向效率最高，沿正交方向也允许但代价很高。
- 在 HL55 的库上，奇数的金属层是横向（M1，M3，M5，T4M2）。
- 在 HL55 的库上，偶数的金属层是纵向（M2，M4，M6）。

立志成才 报国裕民

金属栈

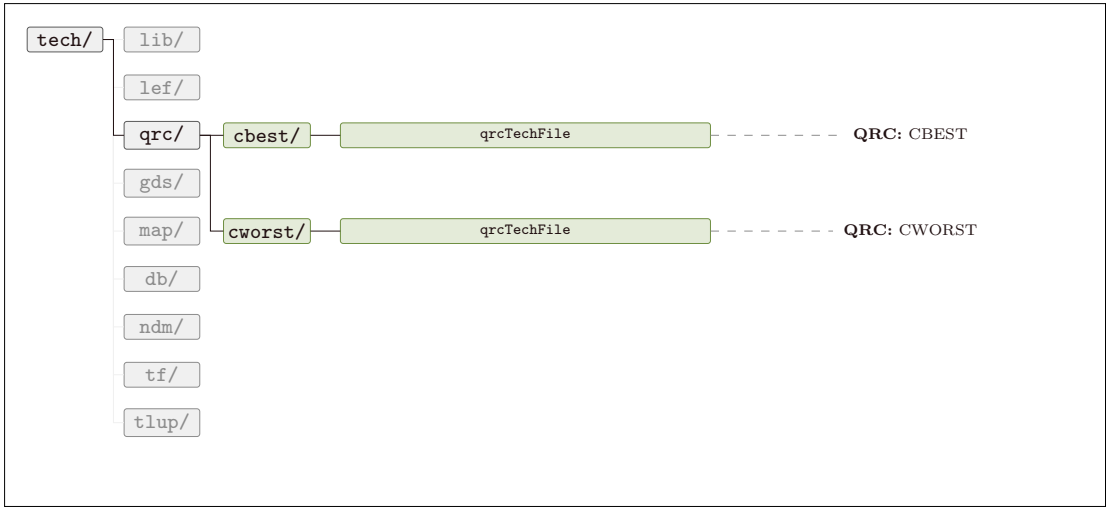

OVERLAP OVERLAP
Overlap
ROUTING RDL V/3.000
Redistribution
Via under RDL
CUT RV
ROUTING T4M2 H/0.400
Thick Metal (4× on minimum line width)
Via under T4M2
CUT T4V2
ROUTING M6 V/0.100
Metal 6
Via over M5
CUT V5
ROUTING M5 H/0.100
Metal 5
Via over M4
CUT V4
ROUTING M4 V/0.100
Metal 4
Via over M3
CUT V3
ROUTING M3 H/0.100
Metal 3
Via over M2
CUT V2
ROUTING M2 V/0.100
Metal 2
Via over M1
Metal Layers
CUT V1
ROUTING M1 H/0.090
Metal 1
Contact to M1
CUT CT
Device Layers
MASTERSLICE PO
Polysilicon (for gate)
IMPLANT LVTP/LVTN/HVTP/HVTN
Implant (for channel)

我们会惊喜地发现这里只需要定义 TECH 的字段，前面 STD 和 MEM 的字段都可以复用！

```
set N_TECH(NAME)      HL055P_7M6X1T4_RDL
```

```
set LEF_TECH          [file join ${LEF_DIR} tech ${N_TECH(NAME)}.lef]
set LEF_STD_RVT       [file join ${LEF_DIR} std ${N_STD(RVT)}.lef]
set LEF_STD_LVT       [file join ${LEF_DIR} std ${N_STD(LVT)}.lef]
set LEF_MEM_M1024     [file join ${LEF_DIR} mem ${N_MEM(M1024)}.plef]
set LEF_MEM_M0256     [file join ${LEF_DIR} mem ${N_MEM(M0256)}.plef]
set LEF_LIST          [list ${LEF_TECH} ${LEF_STD_RVT} ${LEF_STD_LVT} \
                            ${LEF_MEM_M1024} ${LEF_MEM_M0256}]
```

QRC：描述互连金属线的寄生抽取。


tech/
lib/
lef/
qrc/
cbest/
qrcTechFile
QRC: CBEST
gds/
map/
cworst/
qrcTechFile
QRC: CWORST
db/
ndm/
tf/
tlup/

RC 寄生

金属线本身也有延时！有时两个单元间的金属互连线带来的延时比单元本身还大！

金属线的延时来自于其 RC 寄生，由于制造上的偏差，寄生的电容和电阻可能会偏大或偏小

表: RC 角

电阻	电容	名称	速度
电阻偏大	电容偏大	`rcworst`	最慢
–	电容偏大	`cworst`	较慢
–	–	`ctyp`	正常
–	电容偏小	`cbest`	较快
电阻偏小	电容偏小	`rcbest`	最快

该课程的工艺库仅提供了较快的 `cbest` 和较慢的 `cworst` 的 RC 角。

RC 寄生

QRC 文件和 LIB 文件的名称中都包含了 `cbest` 和 `cworst`，这是为什么？

- **互连侧**：QRC 名称中的 RC 角，描述的是单元之间的金属互连线的寄生。
- **单元侧**：LIB 名称中的 RC 角，描述的是单元内部的寄生（HL55 该处只提供了电容偏移）。
- 通常在分析时序时，库文件的搭配应确保单元侧和互连侧使用匹配的 RC 角。

立志成才 报国裕民

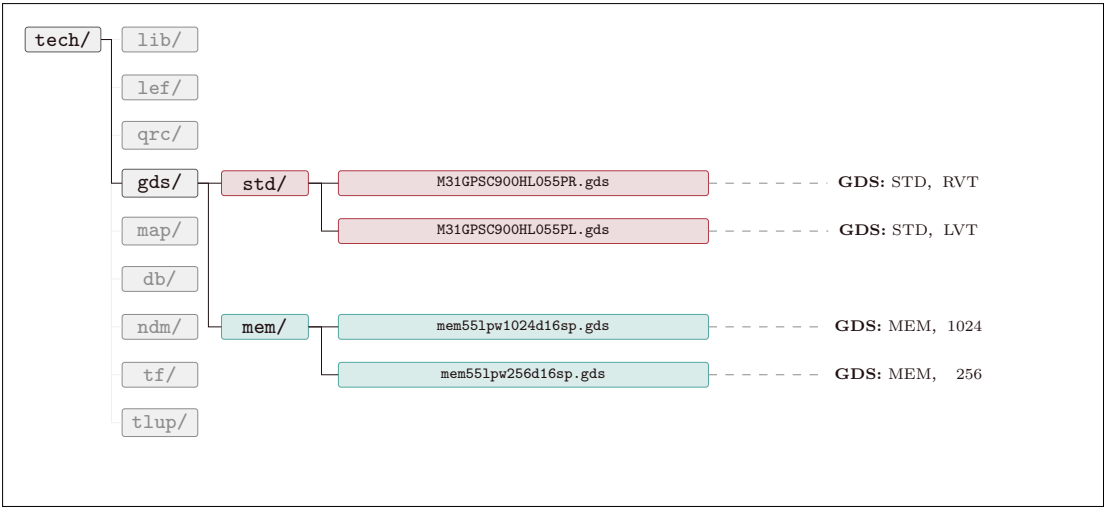
tech.tcl

```
set N_QRC(NAME)      qrcTechFile
set N_QRC(CCB)       cbest
set N_QRC(CCW)       cworst
```

```
set QRC_CCB          [file join ${QRC_DIR} ${N_QRC(CCB)} ${N_QRC(NAME)}]
set QRC_CCW          [file join ${QRC_DIR} ${N_QRC(CCW)} ${N_QRC(NAME)}]
set QRC_LIST         [list ${QRC_CCB} ${QRC_CCW}]
```

GDS：描述单元的版图。

tech/gds


tech/
lib/
lef/
qrc/
gds/
std/
M31GPSC900HL055PR.gds
GDS: STD, RVT
M31GPSC900HL055PL.gds
GDS: STD, LVT
map/
db/
ndm/
mem/
mem55lpw1024d16sp.gds
GDS: MEM, 1024
tf/
mem55lpw256d16sp.gds
GDS: MEM, 256
tlup/

GDS 和 LEF 的关系

GDS 是单元的完整版图，它存储了单元在各个工艺层上的所有内部几何信息

- LEF 可视为 GDS 的抽象，只包含对布局布线有用的几何信息，例如单元的外形和引脚。
- LEF 是文本文件，是可读的，GDS 是二进制文件，且体积要大很多。

P & R 的最终产物就是设计的 GDS 版图：由各单元的 GDS 与设计的互连线按位置合并而来。

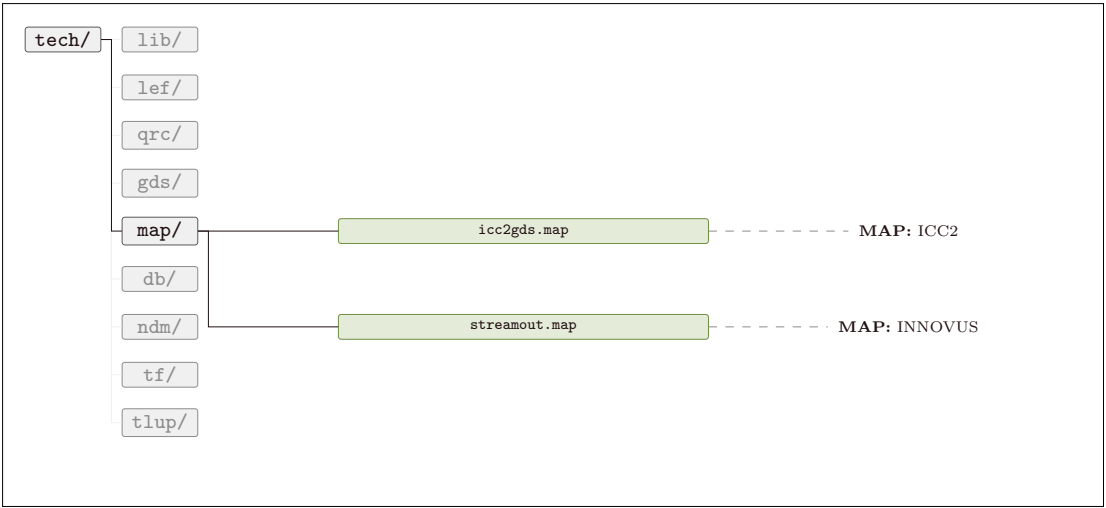
立志成才 报国裕民

tech.tcl


```
set GDS_STD_RVT      [file join ${GDS_DIR} std ${N_STD(RVT)}.gds]
set GDS_STD_LVT      [file join ${GDS_DIR} std ${N_STD(LVT)}.gds]
set GDS_MEM_M1024    [file join ${GDS_DIR} mem ${N_MEM(M1024)}.gds]
set GDS_MEM_M0256    [file join ${GDS_DIR} mem ${N_MEM(M0256)}.gds]
set GDS_LIST         [list ${GDS_STD_RVT} ${GDS_STD_LVT} \
                           ${GDS_MEM_M1024} ${GDS_MEM_M0256}]
```

MAP：描述层号映射。

tech/map


tech/
lib/
lef/
qrc/
gds/
map/
db/
ndm/
tf/
tlup/
icc2gds.map
MAP: ICC2
streamout.map
MAP: INNOVUS

MAP 的作用

MAP 负责将 LEF 中的层名翻译为 GDS 中的层号，本质上是金属层的一张翻译表

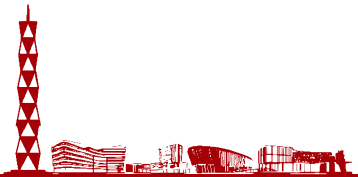
- MAP 文件会在导出 GDS 时使用，负责将 `M1` 这样的层名翻译为 `101` 这样的层号。
- MAP 文件对于 ICC2 和 Innovus 有不同格式，是两个不同的文件。

tech.tcl

```
set N_MAP(CADEN)      streamout
set N_MAP(SYNOP)      icc2gds
```

```
set MAP_FILE_CADEN    [file join ${MAP_DIR} ${N_MAP(CADEN)}.map]
set MAP_FILE_SYNOP    [file join ${MAP_DIR} ${N_MAP(SYNOP)}.map]
```

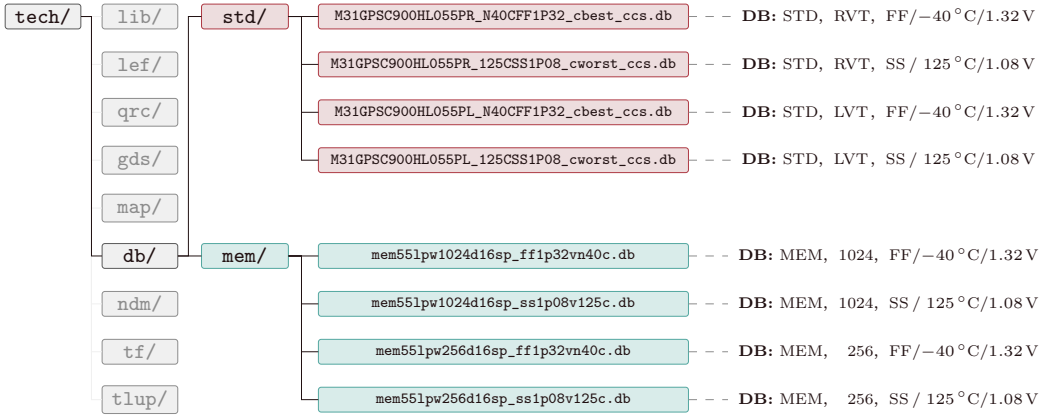
DB：LIB 的二进制格式，Synopsys 工具专用。



DB 实质是 LIB 的二进制版本

- DB 文件是 Synopsys DC 的专用格式，它实质是将 LIB 文件编译为二进制的产物。
- DB 文件可用 LC（Library Compiler）编译，课程库已经提供编译产物。

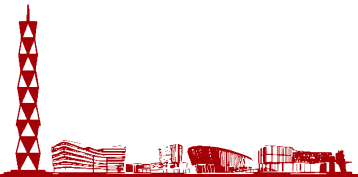
立志成才 报国裕民


tech/
lib/
lef/
qrc/
gds/
map/
db/
ndm/
tf/
tlup/
std/
mem/
M31GPSC900HL055PR_N40CFF1P32_cbest_ccs.db
M31GPSC900HL055PR_125CSS1P08_cworst_ccs.db
M31GPSC900HL055PL_N40CFF1P32_cbest_ccs.db
M31GPSC900HL055PL_125CSS1P08_cworst_ccs.db
mem55lpw1024d16sp_ff1p32vn40c.db
mem55lpw1024d16sp_ss1p08v125c.db
mem55lpw256d16sp_ff1p32vn40c.db
mem55lpw256d16sp_ss1p08v125c.db
DB: STD, RVT, FF/−40 °C/1.32 V
DB: STD, RVT, SS / 125 °C/1.08 V
DB: STD, LVT, FF/−40 °C/1.32 V
DB: STD, LVT, SS / 125 °C/1.08 V
DB: MEM, 1024, FF/−40 °C/1.32 V
DB: MEM, 1024, SS / 125 °C/1.08 V
DB: MEM, 256, FF/−40 °C/1.32 V
DB: MEM, 256, SS / 125 °C/1.08 V

tech.tcl

```
set DB_STD_RVT_FF    [file join ${DB_DIR} std ${N_STD(RVT)}_${N_STD(FF)}.db]
set DB_STD_RVT_SS    [file join ${DB_DIR} std ${N_STD(RVT)}_${N_STD(SS)}.db]
set DB_STD_LVT_FF    [file join ${DB_DIR} std ${N_STD(LVT)}_${N_STD(FF)}.db]
set DB_STD_LVT_SS    [file join ${DB_DIR} std ${N_STD(LVT)}_${N_STD(SS)}.db]
set DB_MEM_M1024_FF  [file join ${DB_DIR} mem ${N_MEM(M1024)}_${N_MEM(FF)}.db]
set DB_MEM_M1024_SS  [file join ${DB_DIR} mem ${N_MEM(M1024)}_${N_MEM(SS)}.db]
set DB_MEM_M0256_FF  [file join ${DB_DIR} mem ${N_MEM(M0256)}_${N_MEM(FF)}.db]
set DB_MEM_M0256_SS  [file join ${DB_DIR} mem ${N_MEM(M0256)}_${N_MEM(SS)}.db]
set DB_LIST_FF       [list ${DB_STD_RVT_FF} ${DB_STD_LVT_FF} \
                           ${DB_MEM_M1024_FF} ${DB_MEM_M0256_FF}]
set DB_LIST_SS       [list ${DB_STD_RVT_SS} ${DB_STD_LVT_SS} \
                           ${DB_MEM_M1024_SS} ${DB_MEM_M0256_SS}]
set DB_LIST          [concat ${DB_LIST_FF} ${DB_LIST_SS}]
```

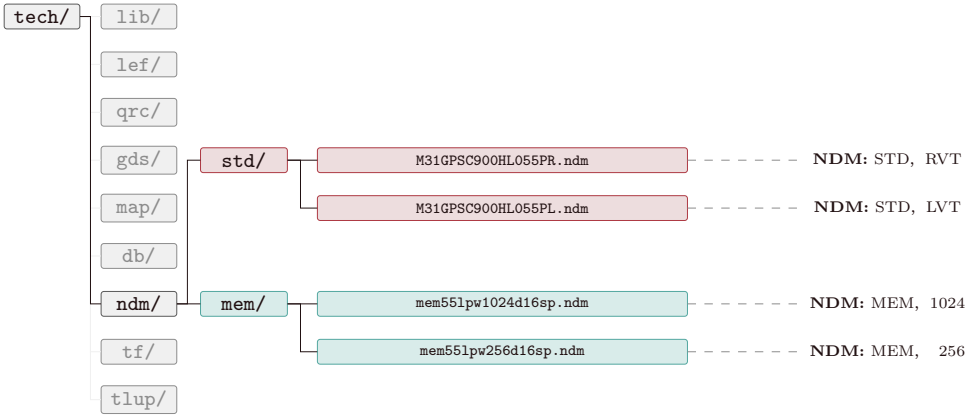
NDM：LEF 和 DB 的封装格式，Synopsys 工具专用。



NDM 实质是 LEF 和 DB 的封装

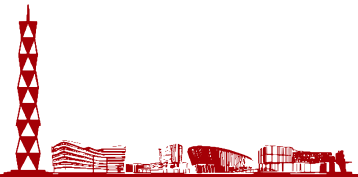
- NDM 文件是 Synopsys ICC2 的专用格式，它实质是将 LEF 和 DB 封装在一起的产物。
- NDM 文件可用 LM（Library Manager）编译，课程库已经提供编译产物。

立志成才 报国裕民

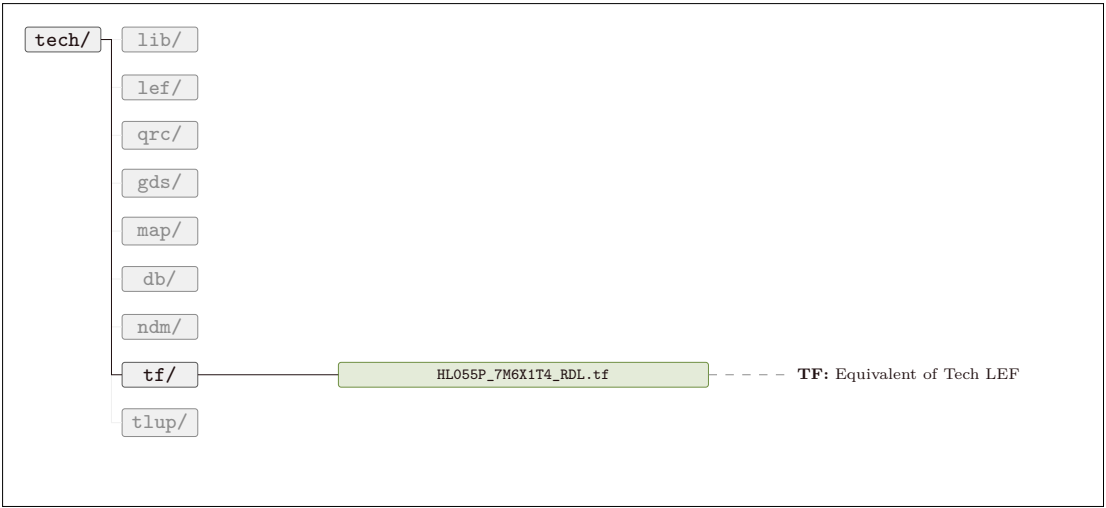

tech/
lib/
lef/
qrc/
gds/
map/
db/
ndm/
tf/
tlup/
std/
mem/
M31GPSC900HL055PR.ndm
M31GPSC900HL055PL.ndm
mem55lpw1024d16sp.ndm
mem55lpw256d16sp.ndm
NDM: STD, RVT
NDM: STD, LVT
NDM: MEM, 1024
NDM: MEM, 256

```
set NDM_STD_RVT      [file join ${NDM_DIR} std ${N_STD(RVT)}.ndm]
set NDM_STD_LVT      [file join ${NDM_DIR} std ${N_STD(LVT)}.ndm]
set NDM_MEM_M1024    [file join ${NDM_DIR} mem ${N_MEM(M1024)}.ndm]
set NDM_MEM_M0256    [file join ${NDM_DIR} mem ${N_MEM(M0256)}.ndm]
set NDM_LIST         [list ${NDM_STD_RVT} ${NDM_STD_LVT} \
                           ${NDM_MEM_M1024} ${NDM_MEM_M0256}]
```

TF：Tech LEF 的等价物，Synopsys 工具专用。



tech/tf


tech/
lib/
lef/
qrc/
gds/
map/
db/
ndm/
tf/
tlup/
HL055P_7M6X1T4_RDL.tf
TF: Equivalent of Tech LEF

TF 实质是工艺 LEF 的另一格式

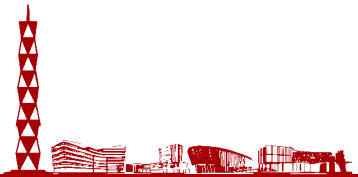
- TF 文件是 Synopsys ICC2 的专用格式，它的作用和 Tech LEF 对等，只是采用不同格式。
- Innovus 用 `HL055P_7M6X1T4_RDL.lef`，ICC2 用 `HL055P_7M6X1T4_RDL.tf`。

立志成才 报国裕民

tech.tcl

```
set TF_TECH           [file join ${TF_DIR} ${N_TECH(NAME)}.tf]
set TF_LIST           [list ${TF_TECH}]
```

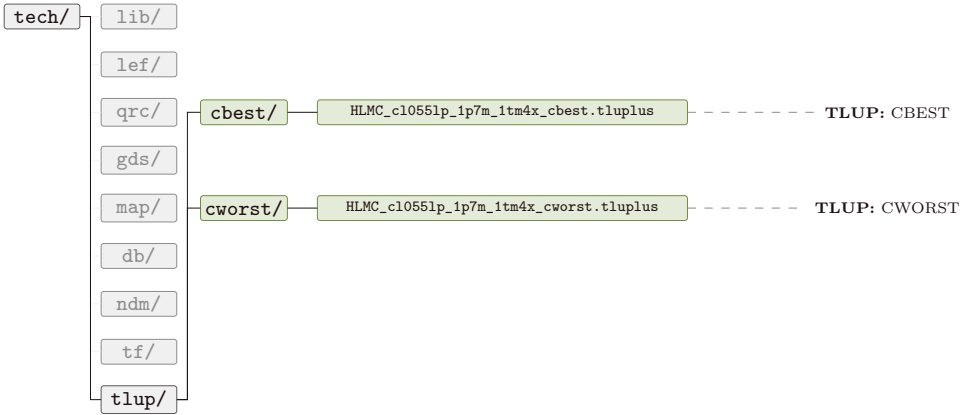
TLUP：QRC 的等价物，Synopsys 工具专用。



TLUP 和 QRC 是两种格式不同的互连寄生文件

- TLUP 和 QRC 都是描述互连寄生的库文件，TLUP 基于二维查表，QRC 基于三维场求解。
- TLUP 用于 Synopsys 工具（通过 `grdgenxo` 由 ITF 生成），QRC 用于 Cadence 工具。

立志成才 报国裕民


tech/
lib/
lef/
qrc/
gds/
map/
db/
ndm/
tf/
tlup/
cbest/
HLMC_cl055lp_1p7m_1tm4x_cbest.tluplus
TLUP: CBEST
cworst/
HLMC_cl055lp_1p7m_1tm4x_cworst.tluplus
TLUP: CWORST

tech.tcl

```
set N_TLUP(NAME)      HLMC_cl055lp_1p7m_1tm4x
set N_TLUP(CCB)       cbest
set N_TLUP(CCW)       cworst
```

```
set TLUP_CCB          [file join ${TLUP_DIR} ${N_TLUP(CCB)} \
                           ${N_TLUP(NAME)}_${N_TLUP(CCB)}.tluplus]
set TLUP_CCW          [file join ${TLUP_DIR} ${N_TLUP(CCW)} \
                           ${N_TLUP(NAME)}_${N_TLUP(CCW)}.tluplus]
set TLUP_LIST         [list ${TLUP_CCB} ${TLUP_CCW}]
```

最后，请回到 tech.tcl 的开头，定义 PVT 角的温度和电压，定义电源的名称。

```
set TEMP_H          125
set TEMP_L          -40
set VOLT_H          1.32
set VOLT_M          1.20
set VOLT_L          1.08
set PWR_NET         VDD
set GND_NET         VSS
set PG_NETS         [list ${PWR_NET} ${GND_NET}]
```

Thank You

谢谢观看