本文摘自《数字[集成电路](https://www.eefocus.com/baike/480353)低功耗物理实现技术与 UPF》 孙轶群  sun.yiqun@nationz.com.cn  国民技术股份有限公司 Nationz Technologies Inc

**4.0 低功耗设计实现实例**  
这里提供一个 DEMO（pl8051\_extend\_chip），可以使读者更快的理解低功耗设计的基础。下图所示，是该设计的功能结构图：

图示

描述已自动生成  
图 4.1 pl8051\_extend\_chip 功能结构图

由图 41 可以看出，整个设计主要是由一个 8051 控制器，通过 SFR 总线对其他模块进行控制，U\_Des 是一个算法模块，U\_Pcu 是功耗控制单元（Power Control Unit）。整个设计都在 Clk 的控制下进行工作，只有 U\_Des 工作在 ClkF 下。

这里假设 Clk（28ns）是 ClkF（7ns）的 4 分频时钟，且与 ClkF 同源同向，这样才可以保证 U\_Des 控制的正确性。

由于 U\_Des 需要在较快的频率下工作，因此需要对 U\_Des 进行一些特殊的设计，也就是对整个设计进行低功耗设计，如下所示：

图示

描述已自动生成  
图 4.2 低功耗设计方案示例

上图主要描述以下信息：  
1.     2 个电压域，TOP 以及 DES\_DOMAIN。  
2.     电源设计外部提供，分 1.2V 的 VDD12 和 1.0V 的 VDD10  
3.     设计中除 DES 电路几乎所有电路工作在 1.0V 的 VDD 电压域。  
4.     U\_Des 由于工作速度要求较高，工作在 1.2V 的 VDD12 电压域

DES\_DOMAIN，但由于不使用的时候需要关断，以降低静态功耗，因此，通过一组 PowerSwitch 进行控制，控制信号来源为 U\_Pcu 的 PcuSfrDatOut[0]输出。该电压域与 TOP 电压域共用 VSS 地信号。

5.     由于 U\_Des 电路处于 VDD12 电压域，而 U\_Pcu 处于 VDD 电压域，因此需要添加 Level Shifter 进行电平转换。

6.     U\_Des 电路输出信号，需要通过一个 Isolation Cell，在关断电源时提供稳定电平，而该电平为 1.2V，因此还需要利用 Level Shifter 转换成 1.0V 电压域信号，Isolation Cell 的 Enable 信号来自于 U\_Pcu 模块的 PcuSfrDatOut[1]信号。

7.     U\_Des 电路中的[寄存器](https://www.eefocus.com/baike/502591)需要使用 Retention Register，save 和 restore 控制信号分别来自于 U\_Pcu 输出 PcuSfrDatOut[2]和 PcuSfrDatOut[3]经过 Level Shifter 产生的信号。

Synopsys 公司为低功耗设计提供了一整套的解决方案，即 UPF 设计流程。UPF（Unify Power Format）是一种描述功耗设计思想的文件，Synopsys Galaxy Implementation Platform 只需要读取 UPF 文件，就可以将低功耗设计思路实现，而 Synopsys Discovery Verification Platform 读取 UPF 后，可以对低功耗思路以及最终实现的电路进行验证。

接下来，我们利用 UPF 描述该 DEMO 的功耗设计思路。

**4.1    申明电压域以及虚拟电压接口**  
首先申明电压域，并申明 VDD、VDD12 以及 VSS 等电压端口：  
create\_power\_domain TOP         
create\_power\_domain DES\_DOMAIN\_domain –elements U\_Des  
create\_supply\_port VDD  
create\_supply\_port VDD12  
create\_supply\_port VSS  
根据上述指令，工具会识别出如下图所示中画圈的信息，电压域及虚拟电压端口开始出现：

图示

描述已自动生成

图 4.3 电压域及虚拟电压端口信息

需要注意的是电压端口只不过是虚拟存在的，并不是一定在顶层连接。

**4.2    申明电源网络**  
接着申明每个电压域中的电源网络，建议从顶层往底层进行申明，如首先申明 TOP 的 VSS、VDD：  
create\_supply\_net VDD         
create\_supply\_net VSS  
以上语句添加下图中画圈部分：

图示

描述已自动生成

图 4.4 产生顶层电源网络

接着申明 DES\_DOMAIN 的电源网络，但需要注意的是，如果需要插入 Level Shifter，最好是在与其有连接关系的电源域（TOP）同时申请电源线，并连接起来：  
create\_supply\_net VDD12         
create\_supply\_net VDD12 –domain DES\_DOMAIN –reuse #DES\_DOMAIN 的 VDD12 与 TOP 的 VSS 共用一个网络，因此需要 -reuse  
create\_supply\_net VDD12G  
create\_supply\_net VDD12G –domain DES\_DOMAIN -reuse  
create\_supply\_net VSS –domain DES\_DOMAIN –reuse  
create\_supply\_net VDD –domain DES\_DOMAIN –reuse  
以上语句添加下图中画圈部分：

图示

描述已自动生成

图 4.5 子电路的电源网路申明

然后将电源网络和电源端口连接起来：  
connect\_supply\_net VDD12 –port {VDD12}         
connect\_supply\_net VDD –port {VDD}  
connect\_supply\_net VSS–port {VSS}  
以上语句添加下图中画圈部分：

图示

描述已自动生成

图 4.6 连接电源网络和电源端口

最后为每个电源域申明主电源网络来源，主电源就是该电压域普通逻辑工作使用的电源：  
set\_domain\_supply\_net TOP –primary\_power\_net VDD –primary\_ground\_net VSS         
set\_domain\_supply\_net DES\_DOMAIN –primary\_power\_net VDD12G –primary\_ground\_net VSS  
做完这一步，电源网络申明基本完成。

**4.3    添加 Power Switch**  
这添加 Power Switch 主要是 2 步，第一步创建 Power Switch，第二步 map Power Switch。可以参看以下脚本进行：  
# Power Switch         
create\_power\_switch des\_sw  
    -domain DES\_DOMAIN  
    -input\_supply\_port {in VDD12}  
    -output\_supply\_port {out VDD12G}  
    -control\_port {des\_po U\_Pcu/PcuSfrDatOut[0]}  
    -on\_state {state\_on in {!des\_po}}  
    -off\_state {OFF {des\_po}}  
map\_power\_switch des\_sw  
    -domain DES\_DOMAIN  
    -lib\_cell HDRSIHVTD0  
create\_power\_switch 和 map\_power\_switch 在综合时只会检查语法，不会有实际效果，但在综合时将该 UPF 指令读入系统，可以由 DC 输出相应的 UPF 给 Synopsys 的 PR 工具，PR 工具可以自动识别这些指令，到时才会真正添加 Power Switch。2 句命令的结果示意在下图画圈中：

图示

描述已自动生成

图 4.7 添加 Power Switch Cell

**4.4    建立电源状态表格（Power State Table, PST）**  
在 Power Switch 插入以后，需要对电压域的有效电压，以及不同电压域之间的关系进行设置，同时建立电压分布及关断的情景（scenarios），可以用于静态电压分析、仿真等。

建立电源状态表格，可以参看以下脚本：  
add\_port\_state VDD –state {LV 1.0} #VDD 是 1.0V，属于低电压域（LV）状态         
add\_port\_state VDD12 –state {HV 1.2} #VDD12 是 1.2V，属于高电压域（HV）状态  
add\_port\_state des\_sw/out –state {HV 1.2}   #Power Switch 输出是 1.2V，属于高电压域（HV）状态  
                   -state {OFF off}#关闭状态（OFF），无电压（off）  
add\_port\_state VSS –state {ON 0.0} #VSS 处于常开状态（ON），电压 0V  
create\_pst design\_pst –supplies {VDD VDD12 VDD12G} #建立一个 PST，由 3 个电源状态不同组成  
add\_pst\_state design\_work –pst design\_pst –state {LV HV HV} #design\_work 状态下，三组电源都开  
add\_pst\_state stand\_by –pst top\_pst –state {LV HV OFF} #stand\_by 状态下，des\_sw/out 输出电源为关闭状态  
需要注意 add\_pst\_state 的 -state 括号里的描述顺序，必须与 create\_pst 的 -supplies 里一致，选择的状态必须与 add\_port\_state 的 -state 里描述的一致才可以。

这样就建立了下表所示的 PST，design\_pst：

表格

描述已自动生成

由此建立了 PST，分析工具或仿真工具就知道需要对哪些状态进行分析了。

**4.5    插入 Isolation Cell**  
下面需要为 DES\_DOMAIN 输出给 TOP 的输出信号添加 Isolation Cell。脚本如下所示：    
# Isolation Cell         
set\_isolation des\_iso\_out  
    -domain DES\_DOMAIN  
    -isolation\_power\_net VDD12 -isolation\_ground\_net VSS  #隔离后只有 VDD12 供电  
    -clamp\_value 1  #关电后输出逻辑 1  
    -applies\_to outputs #对于所有输出添加  
set\_isolation\_control des\_iso\_out  
    -domain DES\_DOMAIN  
    -isolation\_signal U\_Pcu/PcuSfrDatOut[1]  
    -isolation\_sense high  #控制信号为高时 ISO 有效  
    -location self # Isolation Cell 添加在 DES\_DOMAIN 里面

   
 这样就可以产生如下图所示画圈部分电路：

图示

描述已自动生成

图 4.8 插入 Isolation Cell

**4.6    替换 Retention Register Cell**  
接着需要使用 Retention Register Cell 将 DES\_DOMAIN 中在关断电源后需要保留数据的寄存器替换为 Retention Register Cell，可以使用如下脚本：  
# Retention Register         
set\_retention des\_ret -domain DES\_DOMAIN  
    -retention\_power\_net VDD12G -retention\_ground\_net VSS #retention\_power\_net 是关电的 Power，表示需要保持 VDD12G 电源域的数据  
set\_retention\_control des\_ret -domain DES\_DOMAIN  
    -save\_signal {U\_Pcu/PcuSfrDatOut[2] high}  
    -restore\_signal {U\_Pcu/PcuSfrDatOut[3] low}  
map\_retention\_cell des\_ret  
    -domain DES\_DOMAIN  
-lib\_cell\_type RSDFCSD1  
#RSDFCSD1 是带有异步置位的 Retention Register Type，可以从 lib 文件中 retention\_cell 属性中看到。  
根据以上描述，会按照下图画圈示意图描述电路：

图示, 示意图

描述已自动生成

图 4.9 替换 Retention Register Cell

**4.7    插入 Level Shifters**  
插入 Level Shifters 是根据 PST 的定义进行，根据 PST 定义看来，DES\_DOMAIN 和 TOP 之间需要插入 Level Shifters，其中 DES\_DOMAIN 输出的 pin（Isolation Cell 的输出）需要插入 H->L 的 Level Shifter，而其输入则需要插入 L->H 的 Level Shifter。  
插入 Level Shifter 在 Compile 过程中自动进行，也可以利用 UPF 命令 set\_level\_shifter 定义插入的规则。  
插入 Level Shifter 后产生的电路变化示意图如下图画圈处所示：

图示

描述已自动生成

图 4.10 插入 Level Shifters

**4.8    UPF Demo**  
######################         
## CREATE POWER DOMAIS  
######################  
create\_power\_domain TOP  
create\_power\_domain DES\_DOMAIN -elements U\_Des  
#######################  
## TOPLEVEL CONNECTIONS  
#######################  
#VDD  
create\_supply\_port VDD  
create\_supply\_net VDD -domain TOP  
create\_supply\_net VDD -domain DES\_DOMAIN -reuse  
connect\_supply\_net VDD -port VDD  
#VSS  
create\_supply\_port VSS  
create\_supply\_net VSS -domain TOP  
create\_supply\_net VSS -domain DES\_DOMAIN -reuse  
connect\_supply\_net VSS -port VSS  
#VDD12  
create\_supply\_port VDD12  
create\_supply\_net VDD12 -domain TOP  
create\_supply\_net VDD12 -domain DES\_DOMAIN -reuse  
connect\_supply\_net VDD12 -port VDD12  
#VDD12G  
create\_supply\_net VDD12G -domain DES\_DOMAIN  
create\_supply\_net VDD12G -domain TOP -reuse  
#####################  
## PRIMARY POWER NETS  
#####################  
set\_domain\_supply\_net TOP   -primary\_power\_net VDD   -primary\_ground\_net VSS  
set\_domain\_supply\_net DES\_DOMAIN  -primary\_power\_net VDD12G -primary\_ground\_net VSS  
###################  
## DES\_DOMAIN SETUP  
###################  
# Power Switch  
create\_power\_switch des\_sw  
    -domain DES\_DOMAIN  
    -input\_supply\_port {in VDD12}  
    -output\_supply\_port {out VDD12G}  
    -control\_port {des\_po U\_Pcu/PcuSfrDatOut[0]}  
    -on\_state {state\_on in {!des\_po}}  
    -off\_state {OFF {des\_po}}  
map\_power\_switch des\_sw  
    -domain DES\_DOMAIN  
    -lib\_cell HDRSIHVTD0  
# Isolation Cell  
set\_isolation des\_iso\_out  
    -domain DES\_DOMAIN  
    -isolation\_power\_net VDD12 -isolation\_ground\_net VSS  
    -clamp\_value 1  
    -applies\_to outputs  
set\_isolation\_control des\_iso\_out  
    -domain DES\_DOMAIN  
    -isolation\_signal U\_Pcu/PcuSfrDatOut[1]  
    -isolation\_sense high  
    -location self  
# Retention Register  
set\_retention des\_ret -domain DES\_DOMAIN  
    -retention\_power\_net VDD12G -retention\_ground\_net VSS  
set\_retention\_control des\_ret -domain DES\_DOMAIN  
    -save\_signal {U\_Pcu/PcuSfrDatOut[2] high}  
    -restore\_signal {U\_Pcu/PcuSfrDatOut[3] low}  
map\_retention\_cell des\_ret  
    -domain DES\_DOMAIN  
    -lib\_cell\_type RSDFCRD1  
# Level Shifter  
set\_level\_shifter des\_ls\_lh  
    -domain DES\_DOMAIN  
    -applies\_to inputs  
    -threshold 0.1  
    -rule low\_to\_high  
    -location parent  
set\_level\_shifter des\_ls\_hl  
    -domain DES\_DOMAIN  
    -applies\_to outputs  
    -threshold 0.1  
    -rule high\_to\_low  
    -location parent  
#####################  
# ADD PORT STATE INFO  
#####################  
add\_port\_state VDD -state {LV 0.84}  
add\_port\_state VDD12 -state {HV 1.08}  
add\_port\_state VSS -state {ON 0.0}  
add\_port\_state des\_sw/out -state {HV 1.08}  
                          -state {OFF off}  
#############  
## CREATE PST  
#############  
create\_pst design\_pst -supplies {VDD VDD12 VDD12G VSS}  
add\_pst\_state design\_work -pst design\_pst -state {LV HV HV ON}  
add\_pst\_state stand\_by -pst design\_pst -state {LV HV OFF ON}

4.9    DC UPF Flow Script Demo  
source -e ../scr/synopsys\_dc.setup         
source -e ../scr/ variable.tcl  
set\_clock\_gating\_style -min ${CG\_MIN\_BITWIDTH} -max\_fanout ${CG\_MAX\_FANOUT}  
                             -sequential\_cell ${CG\_SEQ\_CELL}  
                             -positive\_edge\_logic ${CG\_POS\_CELL\_LIST}  
                             -negative\_edge\_logic ${CG\_NEG\_CELL\_LIST}  
                             -control\_point ${CG\_CONTROL\_POINT}  
                             -control\_signal ${CG\_CONTROL\_SIGNAL}  
                             -setup ${CG\_SETUP\_VALUE} -hold ${CG\_HOLD\_VALUE}  
set\_operand\_isolation\_style -logic adaptive  
set\_operand\_isolation\_slack 0.1  
set\_svf ${NETLIST\_PATH}${top}${VER}.svf  
read\_verilog ${RTL\_INCLUDE}  
current\_design $top  
uniquify  
link  
load\_upf ${SCRIPT\_PATH}power.upf  
set auto\_insert\_level\_shifters\_on\_clocks all ;#can be added in backend flow  
set\_voltage 1.08 -object\_list {VDD12 VDD12G}  
set\_voltage 0.84 -object\_list VDD  
set\_voltage 0.0 -object\_list VSS  
source -e -v ${SCRIPT\_PATH}timing.tcl  
#M1~M9 are avaliable, M1 will be used by STD  
#set\_ignored\_layers -min\_routing\_layer M4  
set\_ignored\_layers -max\_routing\_layer M7  
report\_ignored\_layers > ${REPORT\_PATH}${top}\_ignored\_layers.rpt  
set\_clock\_gating\_check -setup ${CG\_SETUP\_CHECK} -hold ${CG\_HOLD\_CHECK} [all\_clocks]  
set\_fix\_multiple\_port\_nets -feedthroughs -outputs -buffer\_constant  
set\_attr -type string tcbn90lphpwc0d70d9\_pg.db:tcbn90lphpwc0d70d9 default\_threshold\_voltage\_group SVT\_LS\_LH  
set\_attr -type string tcbn90lphphvtwc0d70d9\_pg.db:tcbn90lphphvtwc0d70d9 default\_threshold\_voltage\_group HVT\_LP\_LH  
set\_attr -type string tcbn90lphpwc0d7\_pg.db:tcbn90lphpwc0d7 default\_threshold\_voltage\_group SVT\_LV  
set\_attr -type string tcbn90lphphvtwc0d7\_pg.db:tcbn90lphphvtwc0d7 default\_threshold\_voltage\_group HVT\_LV  
set\_attr -type string tcbn90lphpwc\_pg.db:tcbn90lphpwc default\_threshold\_voltage\_group SVT  
set\_attr -type string tcbn90lphphvtwc\_pg.db:tcbn90lphphvtwc default\_threshold\_voltage\_group HVT  
set\_attr -type string tcbn90lphphvtcgwc\_pg.db:tcbn90lphphvtcgwc default\_threshold\_voltage\_group HVT\_CG  
set\_attr -type string tcbn90lphphvtwc0d90d9\_pg.db:tcbn90lphphvtwc0d90d9 default\_threshold\_voltage\_group HVT\_ISO  
set\_attr -type string tcbn90lphphvtwc0d90d7\_pg.db:tcbn90lphphvtwc0d90d7 default\_threshold\_voltage\_group HVT\_LS\_HL  
set\_attr -type string tcbn90lphpwc0d90d7\_pg.db:tcbn90lphpwc0d90d7 default\_threshold\_voltage\_group SVT\_LS\_HL  
check\_mv\_design -verbose  
 compile\_ultra -scan -no\_autoungroup -no\_boundary\_optimization  
change\_names -rules verilog -hierarchy  
write -format ddc -hierarchy -output ${DDC\_PATH}${top}\_noscan${VER}.ddc  
write -format verilog -hierarchy -output ${NETLIST\_PATH}${top}\_noscan${VER}.v  
set write\_sdc\_output\_lumped\_net\_capacitance false  
set write\_sdc\_output\_net\_resistance false  
write\_sdc -version 1.5 ${SDC\_PATH}${top}\_cons${VER}.sdc  
check\_design > ${REPORT\_PATH}${top}\_check\_design\_postcomp.rpt  
report\_area > ${REPORT\_PATH}${top}\_area.rpt  
check\_mv\_design –verbose > ${REPORT\_PATH}${top}\_check\_mv\_design.rpt  
report\_constraint -all > ${REPORT\_PATH}${top}\_all\_vio.rpt  
report\_clock\_gating -gating\_elements >> ${REPORT\_PATH}${top}\_icg.rpt  
report\_operand\_isolation -verbose -isolated > ${REPORT\_PATH}${top}\_opiso.rpt  
report\_threshold\_voltage\_group > ${REPORT\_PATH}${top}\_VT.rpt  
report\_threshold\_voltage\_group -verbose >> ${REPORT\_PATH}${top}\_VT.rpt  
save\_upf ${NETLIST\_PATH}${top}${VER}\_dc.upf #upf’ for next part of flow

5.0  总结  
本文从 [CMOS](https://www.eefocus.com/baike/481845) 电路功耗原理入手，针对不同工艺尺寸下[数字集成电路](https://www.eefocus.com/baike/480369)的低功耗物理实现方法进行描述，并通过一个例子介绍了 Synopsys UPF（Unified Power Format）文件对低功耗设计的描述原理。UPF 是 Synopsys 公司提出的一种对芯片中电源域设计进行约束的文件格式。通过与 UPF 格式匹配的 Liberty 文件，UPF 约束文件可以被整套 Galaxy 物理实现平台的任何一个环节直接使用，并将设计者的电源设计约束传递给设计工具，由工具完成设计的实现工作，从而实现整套数字集成电路低功耗物理实现的流程。