Power Domain

create_power_domain pd1	Power domain created at top level scope
create_power_domain pd2 -elements cpu	specify instance name
create_power_domain pd3 -elements Mem -scope cpu	in relative scope cpu, exists instance Mem

set_domain_supply_net is required to specify primary power and ground nets for a power domain

set_domain_supply_net TOP -primary_power_net VDD -primary_ground_net VSS

Power Switches - MTCMOS Cells

create_power_switch myswitch -domain Core -input_supply_port {in VDDC} -output_supply_port {out VDDS } -control_port {Nsleep PMU/sleep} -on_state {on_state in {!sleep} }

Place & Route later will physically map the constraint to the real switch cells:

map_power_switch -domain <domain_name> -lib_cell <lib_cell_name> <switch_name>

Even when not switching,CMOS cells consume leakage power. The idea is to **save leakage power by truning off design partitions when the logic is inactive**,for that, Power Switches are added between the main supply and the virtual supply to be shutdown. Of course the virtual supply is connected to standard cells supply rails.

Power Supply Network UPF 1.0

create_supply_port VDD	Creates Port VDD
create_supply_net VDD -domain TOP	Creates supply net at top level
connect_supply_net VDD -ports VDD	Connects supply net to port VDD

Power Supply Network UPF 1.0 (cont)

set_domain_supply_net	Specifies primary power/ground nets for power domain

create_supply_net -reuse will create supply net and supply port automatically at child level

Level Shifters - set_level_shifter

set_level_shifter LS_in -domain PD2 **-applies_to inputs -rule high_to_low** -location **self**

set_level_shifter LS_out -domain PD2 **-applies_to outputs -rule low_to_high** -location **parent**

rule can be " high_to_low", "low_to_high" or "both"	location can be "self", "parent" or "automatic"

LS needed when 2 different signals belonging to different voltage power domains need to be connected to ensure proper signal value and timing propagation
Level Shifter strategy is **optional** since insertion controlled by PST.
LS Strategy can be applied to control the**location** of the LS cells

ISO Cells-set_isolation/set_isolation_control

set_isolation -applies_to outputs -location parent

set_isolation –no_isolation –elements {list}

set_isolation iso_core_out -domain Core_domain -isolation_power_net VDDC -isolation_ground_net VSS -clamp_value 1 -applies_to outputs

set_isolation_control iso_core_out –domain Core_domain -isolation_signal PwrCtrl/isolate_ctrl -isolation_sense low -location parent

Provide protection during shutdown to avoid spurious signal propagation

Retention FF

set_retention RET_PD1 -domain PD1 -retention_power_net VDDH -retention_ground_net VSS

set_retention_control RET_PD1 -domain PD1 **-save_signal** {SAVE high} **-restore_signal** {RESTORE high}

map_retention_cell -domain PD1 **-lib**_cells

Retention FF will maintain the state of sequential elements in a shutdown block.
Retention supplies must be "Always-ON" as long as the FF need to hold their value

Power State Table - PST

Once you know more about what you are trying to define, specify the design switching characteristics in a power state table.
This requires knowing (or deciding) the operational voltages for each power domain, and the supplies being used.
The PST defines clearly **all legal voltage states and power state combinations** for a design, hence it becomes the center of the design power intent as it **captures the dynamic voltage scaling (DVS/DVFS) and shutdown scenarios**.

add_port_state <port_name> {-state {name <nom>|<min nom max>|<off>}}
Defines all possible state information to a supply port
create_pst <table_name> -supplies {list}
Creates a PST using a specific order of supply nets
add_pst_state <state_name> -pst <table_name> -state <suppl-y_states>
Defines valid combination of supply net values for each possible state of the design

Create Block Level UPF

To specify driver supply set on input ports and receiver supply set on output port ,one can use **set_port_attributes** command(SPA),

set_port_attributes -driver_supply_set | -receiver_supply_set VDD_set

To specify supply net on ports if no corresponding supply set is defined, use set_related_supply_net(SRSN):

set_related_supply_net –object_list A –power VDD

Idea here is to create boundary power constraints for IO ports of blocks to guide block level implementation.
Also when no SRSN/SPA is applied user cn specify default supply to be used to power ports:
set_port_attributes -elements { . } -attribute related_supply_def-ault_primary TRUE

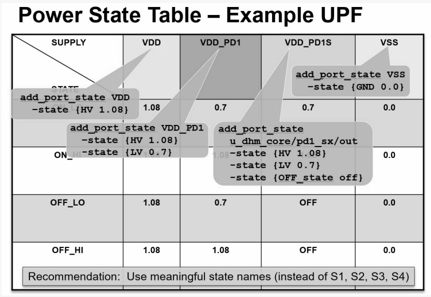
Importance of Power State Table

Recommendation to use the supply_net names for create_pst command to keep PST matrix readable.
Power Management cell insertion is based on PST and UPF constr-aints:
Level shifter cells are inserted based upon the PST
Isolation cells are inserted based upon the UPF constraints
Retention cells are inserted based on constraints
Correctness of the design is checked against PST.

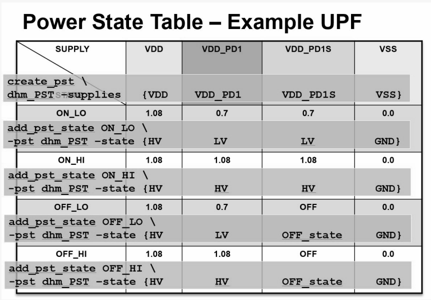
Always-On Logic

Some logic needs to stay active during shutdown
1-Path to enable pins of ISO/ELS
2-Power switches
3-Retention registers
4-Feedthrough paths
Always-on logic remains powered within shutdown block
1-Single-rail AO cells
2-Dual-rail AO cells

PST Example



PST Example



Supply Sets

create_power_domain doesn't require the use of **set_domain_supply_net** as pre-defined handles **primary.power** and **primary.ground** are automatically created.
Hence :
set_isolation doesn't require use of **-isolation_power-_net** or **-isolation_ground_net** options
set_retention doesn't require use of **-rentention_powe-r_net** or **-retention_ground_net** options
set_isolation -source SS_VDD1 **-sink** SS_VDD3 -> using source/sink provides finer control for inserting isolation cells to only on ports where the driver supply is related to a given supply set and load supply is related to another supply set. **Works for only for supply sets**

set_isolation -diff_supply_only true will insert iso cells only for ports where driver and load usea different supply. **Works for both supply sets and supply nets.**

Supply Sets - 2 use models

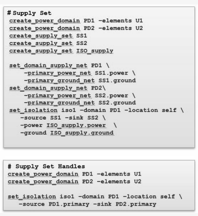
Explicit: **create_supply_set mySS1 -function {power myVDD} -function {ground myVSS}**

Implicit: Supply set handles automatically created with the **create_pow-er_domain** command

Predefined handles are : **PD1.primary PD1.default_isolation PD1.defau-lt_retention**

Supply sets provide an **abstract way of bundling supply nets** and doing the front-end implementation **without knowing the supply net and port names**. Physical net and port information is refined at the back-end. This **helps** improve the **re-usability of your IP since the IP is implemented without the actual nets and ports** and thus is more portable.
Complete supply network information is not needed when you do your front end implementation.

Supply Set Handles-Explicit vs Implicit



```
# Supply Set
create_power_domain PD1 -elements U1
create_power_domain PD2 -elements U2
create_supply_set SS1
create_supply_set SS2
create_supply_set ISO_supply

set_domain_supply_net PD1 \
    -primary_power_net SS1.power \
    -primary_ground_net SS1.ground
set_domain_supply_net PD2\
    -primary_power_net SS2.power \
    -primary_ground_net SS2.ground
set_isolation iso1 -domain PD1 -location self \
   -source SS1 -sink SS2 \
   -power ISO_supply.power  \
   -ground ISO_supply.ground
```

```
# Supply Set Handles
create_power_domain PD1 -elements U1
create_power_domain PD2 -elements U2

set_isolation iso1 -domain PD1 -location self \
   -source PD1.primary -sink PD2.primary
```