



Path-based UPF Strategies: Optimally Manage Power on your Designs

Progyna Khondkar
Siemens EDA

SIEMENS



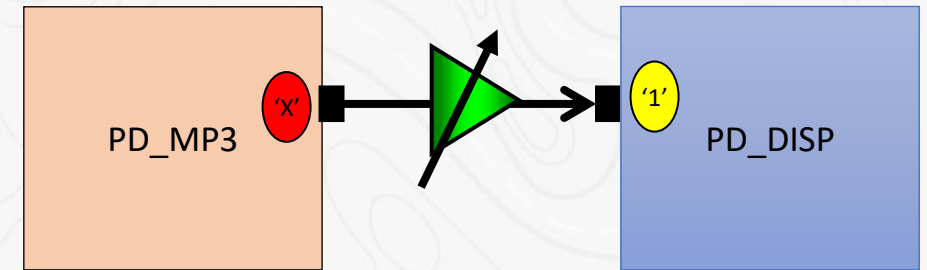
Agenda

- What's Port based and Path based UPF Semantics for Strategies?
- How Path Based Semantics Works?
- Why Port Based Semantic was Ad hoc/Inefficient?
- Path based ways to optimally manage power on any design
- Case Studies/Examples
 - Easy transition from a port-based to path-based standard methodology
- Concluding remarks
- References

Background: Isolation Strategy

- Source logic goes **OFF** and sink logic is in **ON** state, there is a need for Isolation

```
set_isolation strategy_name  
[-source <source_domain_name | source_supply_ref >]  
[-sink <sink_domain_name | sink_supply_ref >]  
[-location <self | other | parent | fanout>]  
[-clamp_value <0 | 1 | Z | latch | value >]
```



- If ISO is **not** inserted properly at right path, then it can lead to **functional failure**
- If ISO is placed at a **location where it is not required**,
 - It is a **redundant - waste** area & power.

Port Vs Path Based Semantics

- UPF 3.0 & 3.1, introduces **path-based** semantics for ISO,LS, RPTR with **-sink & -diff_supply_only TRUE**
 - Explicitly defines paths from source to sink domains for these ISO, LS or RPTR
- Before path-based - ISO, LS, RPTR utilized ad hoc methodologies,
 - Such as **port splitting** – cause difficulties to manage power
e.g. when ISO specified on ports & port fanouts to multiple receiving logic supplies,
 - The **port-based** semantics would place **ISO cells on all paths**.
- **Path-based** semantics place ISO in paths that goes to specified **sinks** or **receiving logic**
 - Split net (**net splitting**) relevant to the port and place **ISO cell that goes to specific sink** only
 - Allows to **optimally** manage power on any design
- But **path-based** methodology adoption - not straight forward
 - Depends on the **contents** of strategies
e.g., ISO with **-location fanout/self/parent** pose extra complexity to imply a strategy according to expectations.

Path Based Semantics

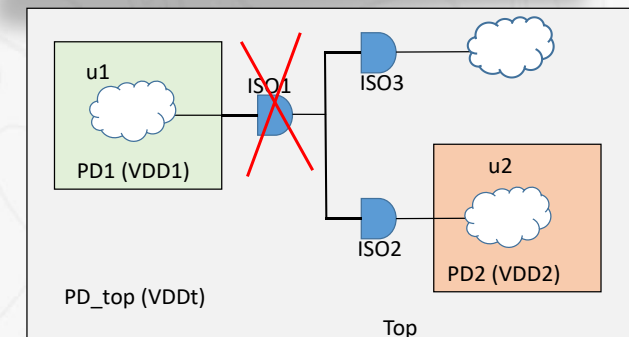
How it Works:

- **ISO1** has lowest precedence
- **ISO2 & ISO3** have same precedence
 - However- covering different sinks
 - Different paths – hence no conflict
- Tool must **drop** ISO1 – due to lower precedence
- Tool must **honor** ISO2 & ISO3
 - Each Case of ISO2, ISO3 – cells must be placed **as close to sink domain possible**
 - **Net Splitting facilitates this placements**

NOTE:

- ISO Strategies are applied on a **per path** basis in the design
- Net splitting helps minimizing **collateral damage(?)**
- For **-location fanout**: Target insertion port must be on **target domain** boundary **closes to receiving logic**
- For **-location self/parent/child**: Target insertion port must be on **target domain** (self or parent or child as applicable)
- Tool must **error** out when its **impossible to infer/insert ISO without port duplication**

```
set_isolation ISO1 –elements {u1}  
set_isolation ISO2 –elements {u1} –sink {VDD2}  
set_isolation ISO3 –elements {u1} –sink {VDDt}
```

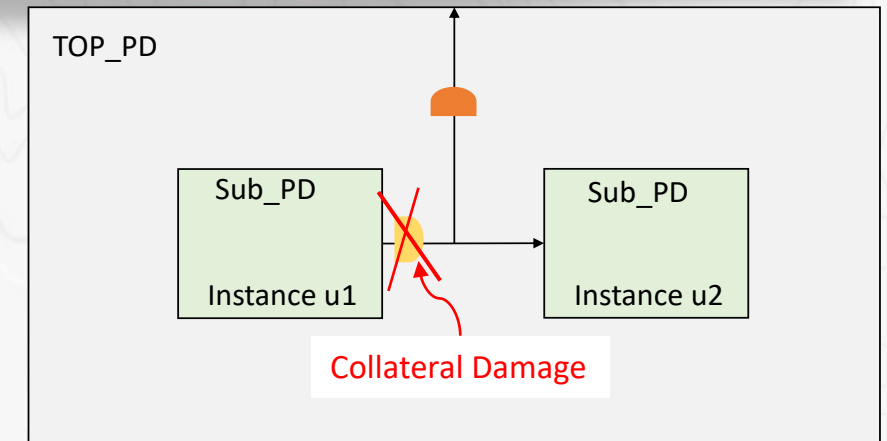


Port Based Semantics

Why its ad hoc:

- Collateral Damage(?) from Port Based Semantics
- Examples (right hand side block diagram).
 - TOP_PD with two sub blocks
 - Instance u1 and u2 in the Sub_PD power domain
 - O/P of instance u1 goes to the top and to instance u2
 - PD Crossing: sub_PD(u1) to TOP_PD
- Based on the ISO strategy – Port-based semantics implies
- Port-splitting will infer/insert two ISO cells for each path
 - However, only one should be sufficient

- ISO strategy
 - domain Sub_PD -source Sub_PD.primary \
-sink TOP_PD.primary



Path Based Semantics

Path based ways to optimally manage power on any design (1)

set_isolation iso1 -domain Blue -sink Green

For **-location self**:

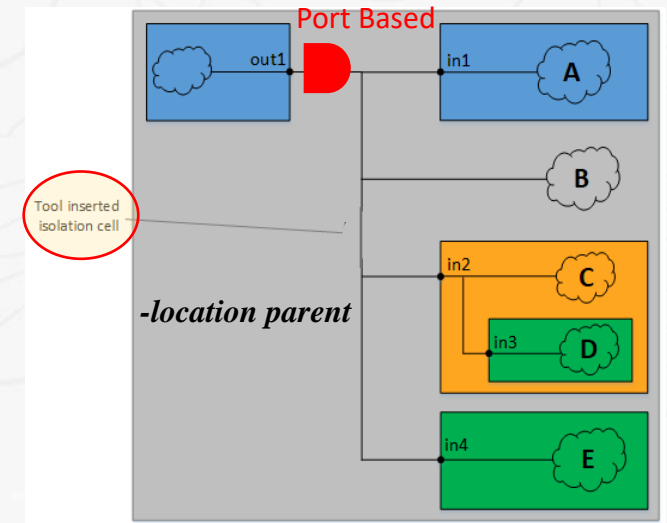
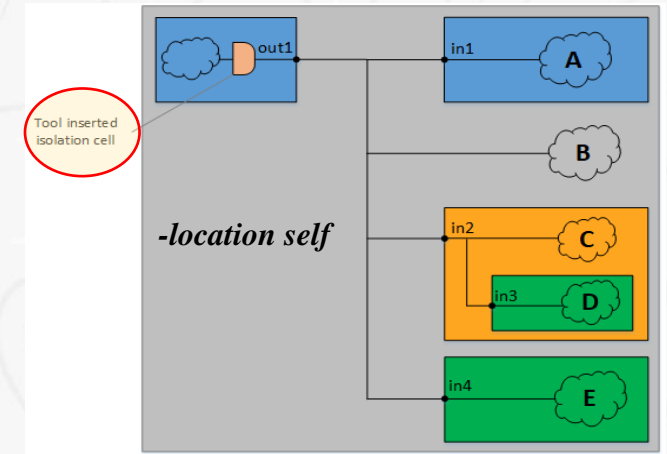
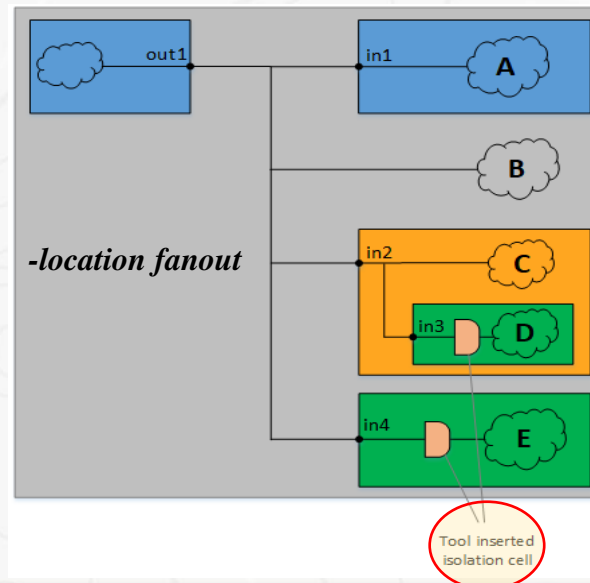
- Cannot implement the ISO that affects only the target receiver (**green domain**).

For **-location parent**:

- Cannot implement the ISO that affects only the target receiver (**green domain**).
- Tool give warning and inserts the ISO in parent domain (**path to the green domain**)

For **-location fanout**:

- Can implement the isolation strategy that affects only the target receivers (**-sink**).



Path Based Semantics *set_isolation iso2 -domain Blue -sink Orange*

Path based ways to optimally manage power on any design (2)

For **-location self**:

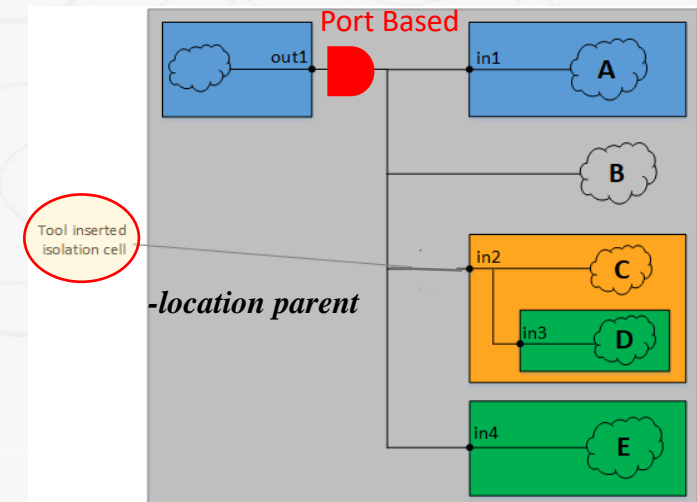
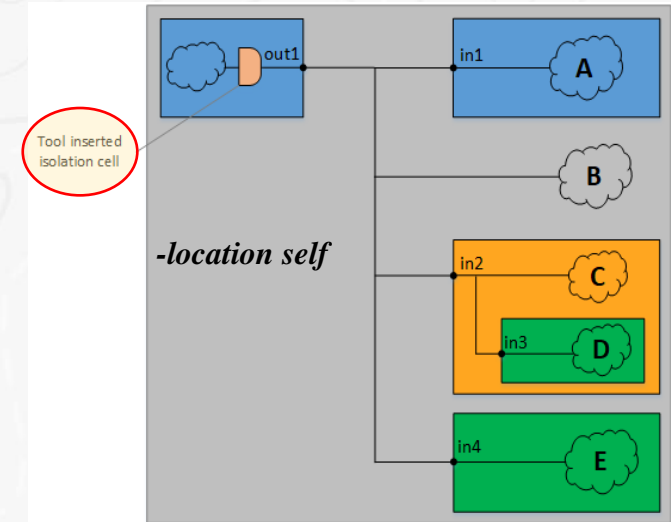
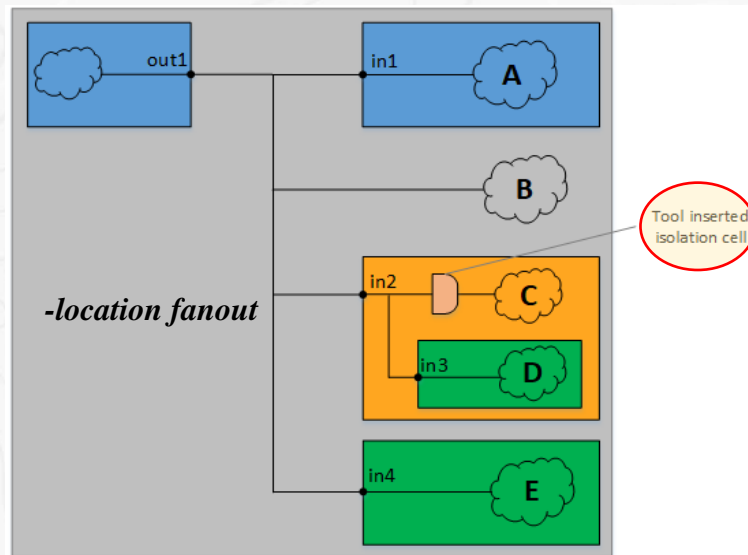
- Cannot implement the ISO that affects only the target receiver (**orange domain**).

For **-location parent**:

- Cannot implement the ISO that affects only the target receiver (**orange domain**).
- Tool give warning and inserts the ISO in parent domain (**path to the green domain**)

For **-location fanout**:

- Cannot implement the isolation strategy that affects only the target receivers (**-sink**)



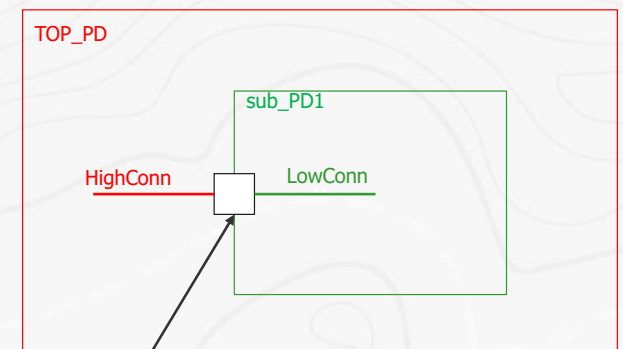
Path Based Semantics

Path based ways to optimally manage power on any design (Summary of Cases)

- It is evident from the case studies –
 - On a real design, **-location self** shows similar placements of ISO cells.
- However, for **-location parent** and **-location fanout**,
 - **Port-based** and **Path-based** semantics differ significantly.
- This difference allows users to comprehend **optimal** and **accurate** designation of
 - ISO cells from a very early stage of the design phase at the **RTL**,
 - Which bounds to comply on down the implementation phases for UPF protection cells insertion
 - Whether it is during **synthesis** or the **place&route** stages

NOTE:

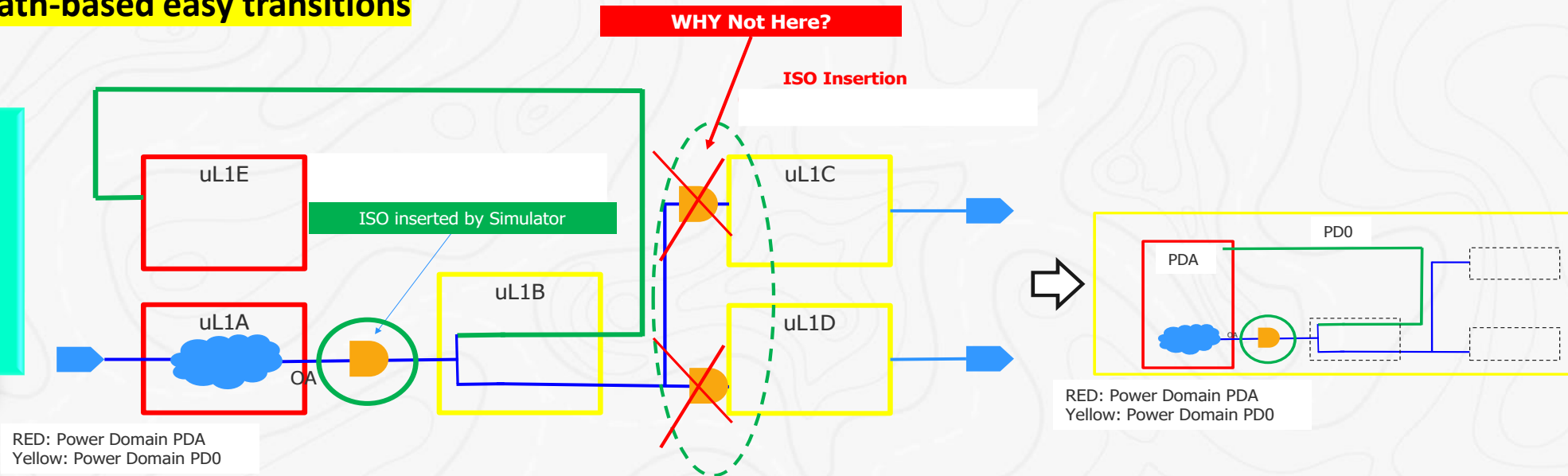
- Its important to understand **Highconn/LowConn** side of Ports
- Since Path based applies ISO on ports at **HighConn** in parent instance
e.g. TOP_PD is parent in Sub_PD1 contexts and shows Highconn of a port



Why ISO not before instance uL1C & uL1D but only before uL1B?

Port-based to path-based easy transitions

```
set_isolation ISO_PDA \  
-domain PDA \  
-isolation_supply_set ss_PDO \  
-clamp_value 0 \  
-applies_to outputs \  
-diff_supply_only TRUE \  
-isolation_signal {uPMU/ISOEN} \  
-isolation_sense low \  
-location parent
```



- **Red** instances are within **PDA** and **Yellow** are within **PDO** domain.
 - When **set_isolation -location parent** is specified on a port,
 - Tool does not insert ISO anywhere in parent instance.
- Tool only considers ports at HighConn in parent instance.
 - Obviously, the implemented algorithm **do not consider** adding ISOs in the **green dotted circle**.
 - As well - tool gives **warning** because port-splitting is a violation.

Warning: dut.upf(41): (vopt-9927) Isolation strategy 'ISO_PDA' specified on port '/TOP/uL1A/OA' can't be applied without splitting the port '/TOP/uL1A/OA'. To avoid port splitting, cell will be inserted for the paths that go to target receiver supply.

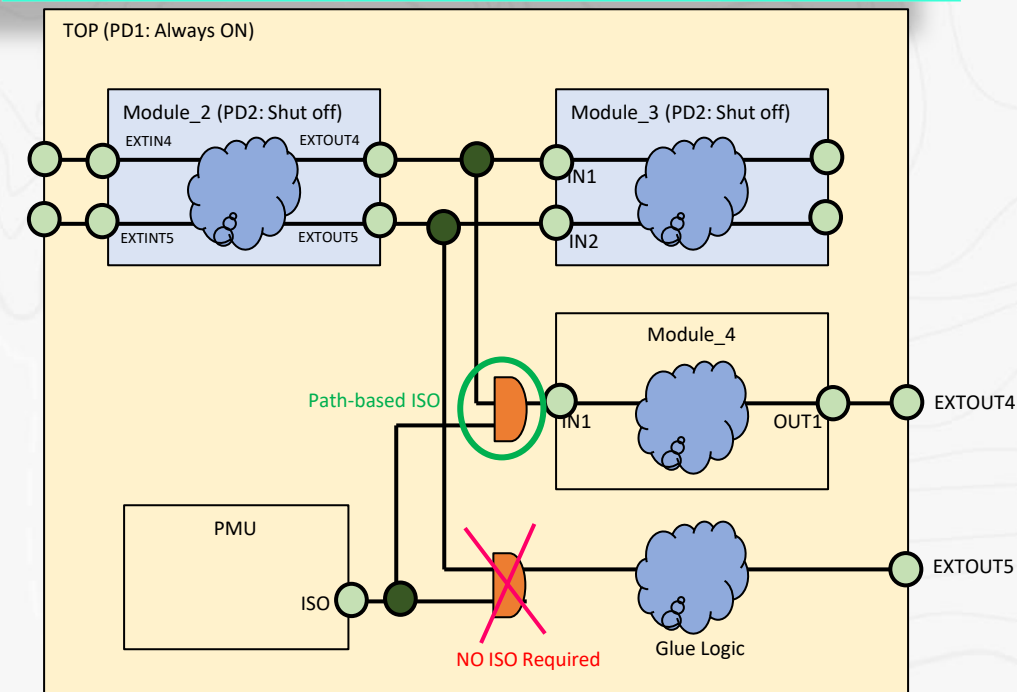
What happens to ISO for glue logic?

Port-based to path-based easy transitions

- Big concerns -ISO placement on glue logic on fanout path?
 - Heterogeneous fanout scenario
 - **Enable/Disable** options to allow/disallow ISO before glue-logic.
- **Module_2** output are not directly connected to glue-logic, not path-based
 - But they are connected to top-level module output ports
 - Only a path going out from the **Module_2**
- So, “tool option to disable ISO before glue-logic”
 - No effect on these ports.
- Such a **disabling ISO** option will take effect only when:
 - Port is directly connected to the glue-logic
 - Multiple paths corresponding ISO strategy is not applicable to one of these paths.
- ISO will be inserted as shown in green circle because for **Module_2**
- Disabled ISO before glue-logic will not insert ISO on glue-logic path
 - If the port has multiple paths and
 - Corresponding ISO strategy is not applicable to at least one of the paths.
- This is shown by **red crossed line** and **NO ISO REQUIRED** message

```
set_isolation_control
ISO_PD2_out_0 \
-domain PD2 \
-isolation_signal {PMU/ISO} \
-isolation_sense low \
-location fanout
```

```
set_isolation ISO_PD2_out_0 \
-domain PD2 \
-isolation_supply_set ss_PD1 \
-clamp_value 0 \
-applies_to outputs \
-diff_supply_only TRUE
```



Concluding Remarks

- UPF protection cells like ISO are required when source logic goes OFF and sink logic is - in ON state
 - If ISO is not inserted properly at right path, then it can lead to *functional failure*.
 - If ISO is placed at a location where it is not required, it is redundant, *waste areas* and *consumes unnecessary power*.
- Path-based semantics allows to **manage power** in *very efficient & optimal way*
 - Over the legacy **port-based** semantics.
- With the **path-based semantic incorporated** - the tool inserts an ISO, LS, RPTR
 - *Closest to*, and *connected to*, the target insertion port within the *extent of the location domain*
- For ISO, LS, RPTR cell strategies — if user specify **-sink** in the strategy,
 - Then the *inserted cell affects receivers* that are powered by a supply set that **matches** *the specified supply set*.
- For ISO - if user specify **-diff_supply_only TRUE** in the strategy,
 - Then the *inserted cell affect receivers* that are powered by a supply set that **does not** *match the driving supply set*.
- So **-sink** and **-source directly manipulate -location**, and UPF 3.1
 - Specifies the path *clearly* on which ISO, LS and REPTR must be applied
- In case if the **semantics** and algorithms **implementation** are not **incorporated** in a tool
 - Then **tools** should **error out with appropriate messaging**

References

- [1] IEEE Std. 1801™-2009 "IEEE Standard for Design and Verification of Low-Power Integrated Circuits", 6 March 2013.
- [2] IEEE Std. 1801™-2013 "IEEE Standard for Design and Verification of Low-Power Integrated Circuits", IEEE Computer Society, 29 May 2013.
- [3] IEEE Std 1801™-2015 "IEEE Standard for Design and Verification of Low Power, Energy Aware Electronic Systems IEEE Computer Society, 05 Dec 2015.
- [4] IEEE Std 1801™-2018 "IEEE Standard for Design and Verification of Low Power, Energy Aware Electronic Systems" IEEE Computer Society, 29 March 2019