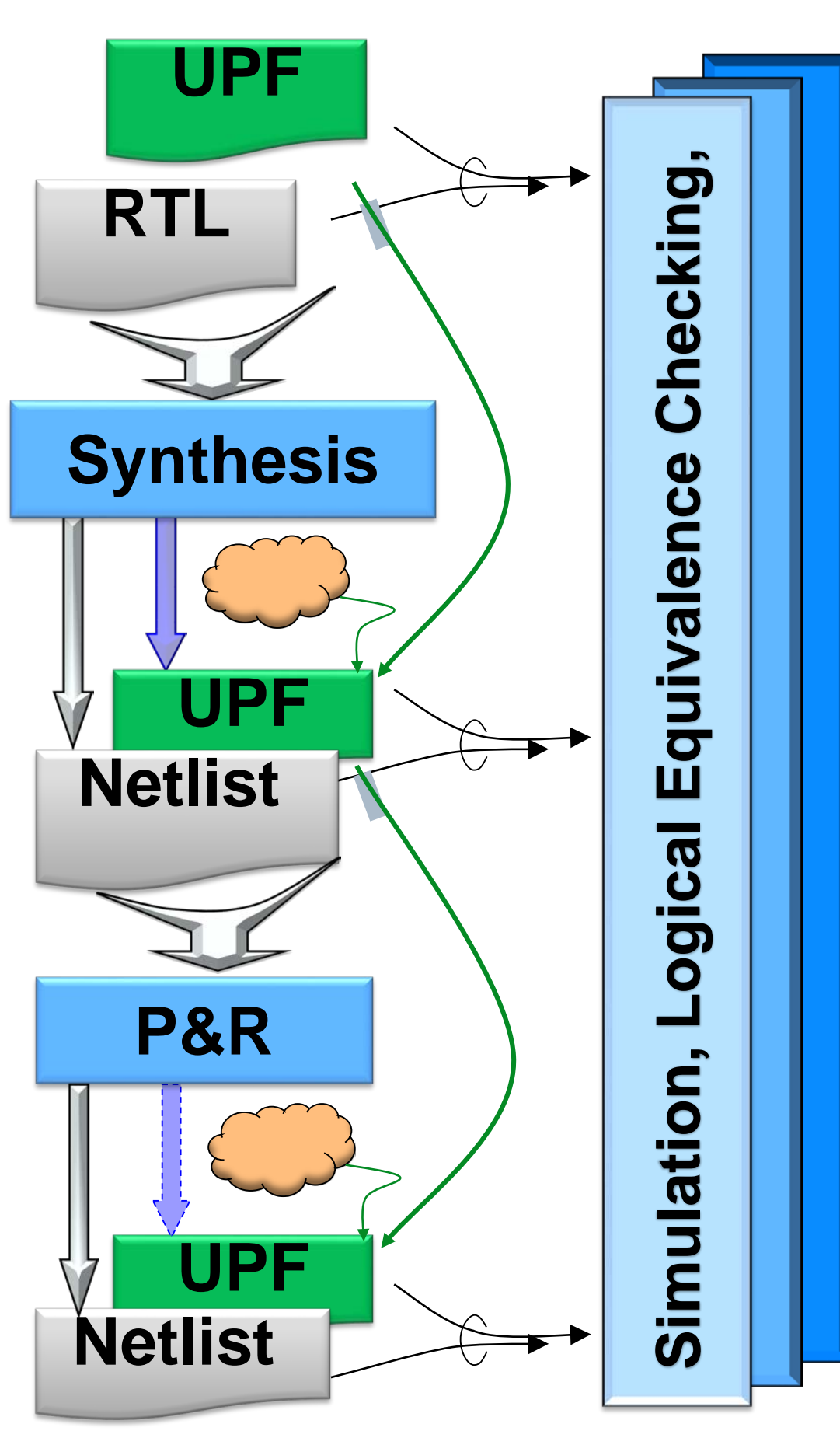
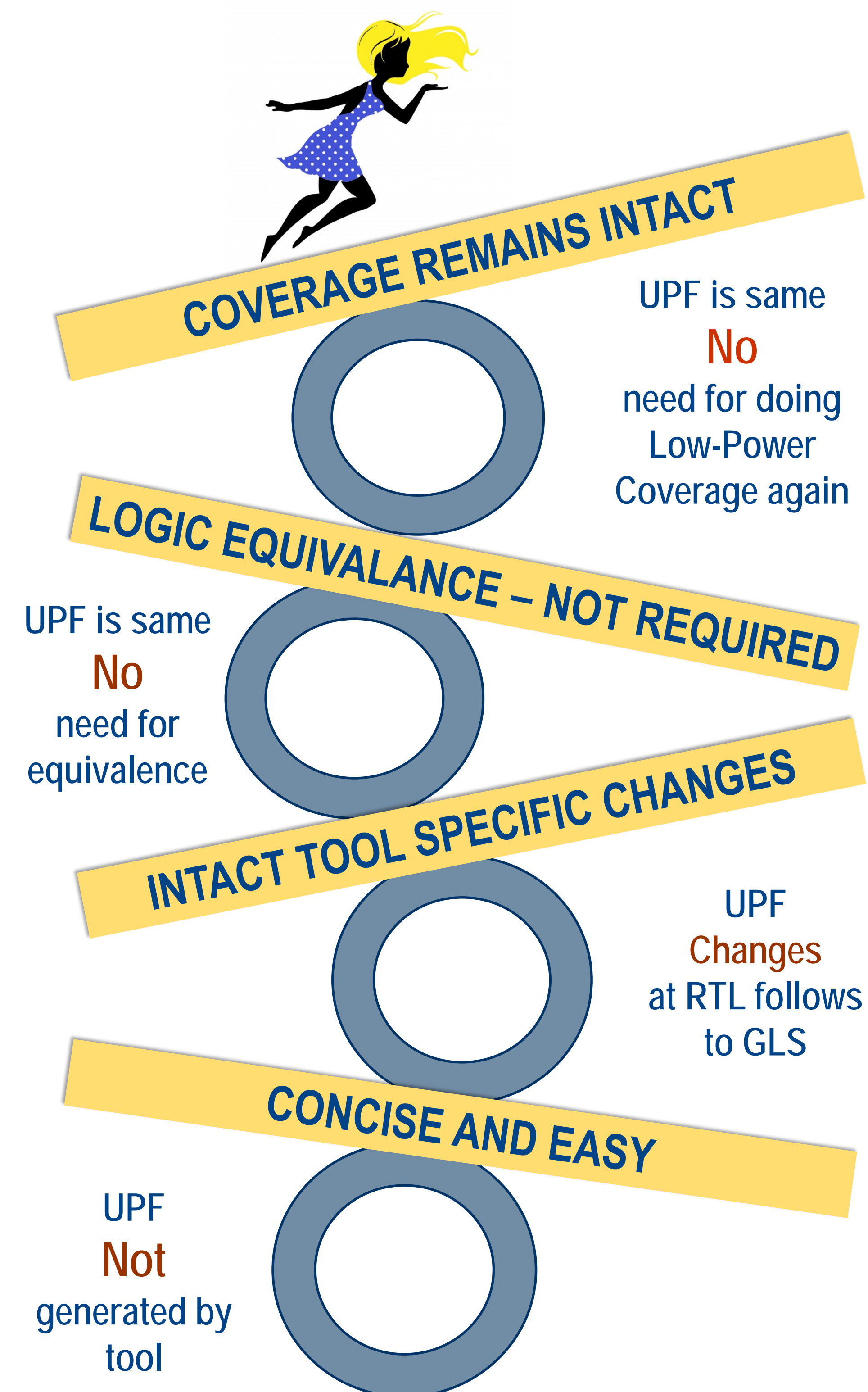


## Abstract

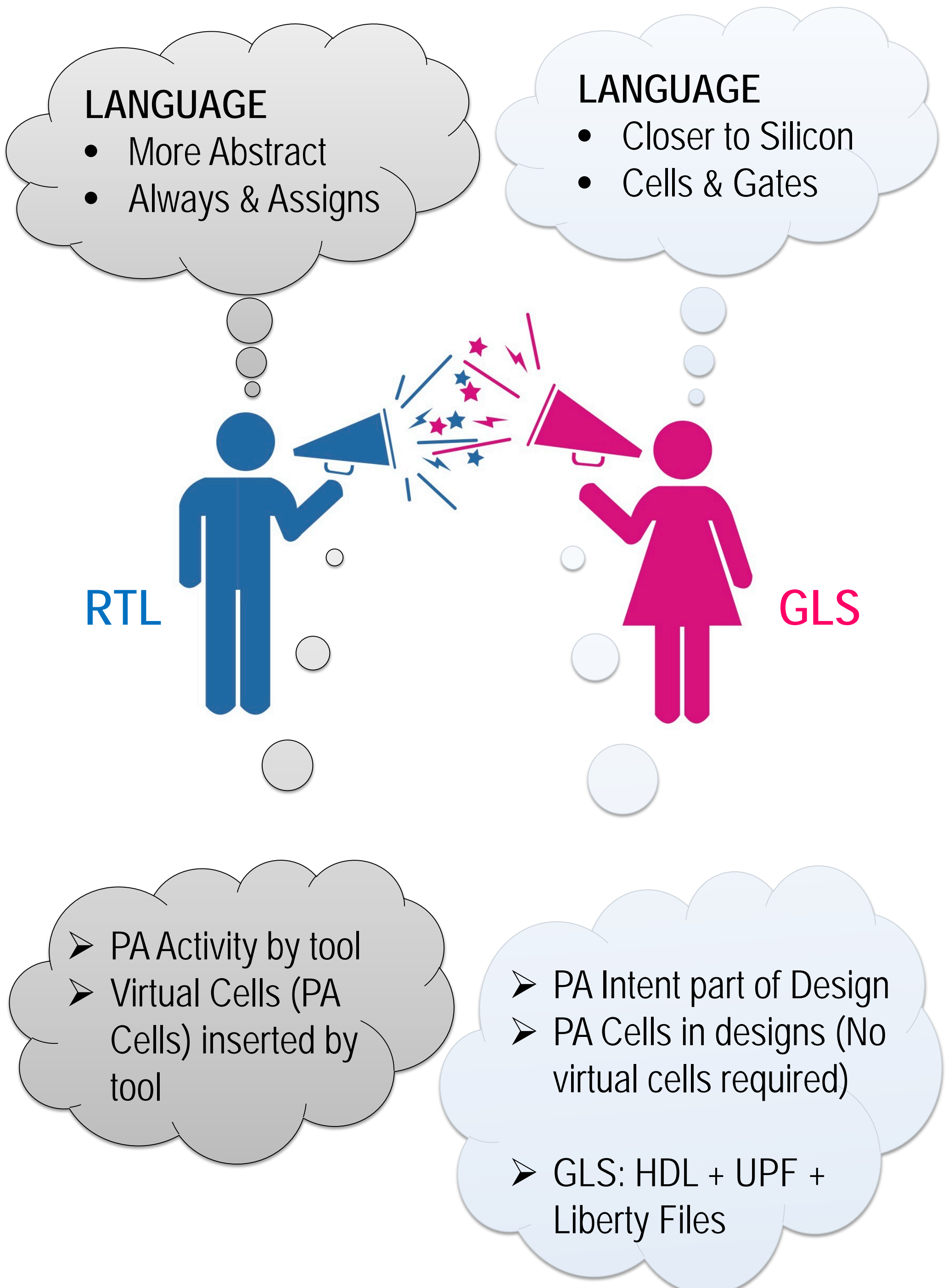
- Often UPF needs to be modified at the next verification stage (RTL to GLS)
  - Hierarchy changes
  - Cell placements
  - Cell connections
- Problems
  - Managing different UPFs
  - Logic Equivalence
- Highlight differences between RTL and GLS UPF
- Proposed methodology to write RTL UPF
- Minimal UPF changes required during gate-level power verification



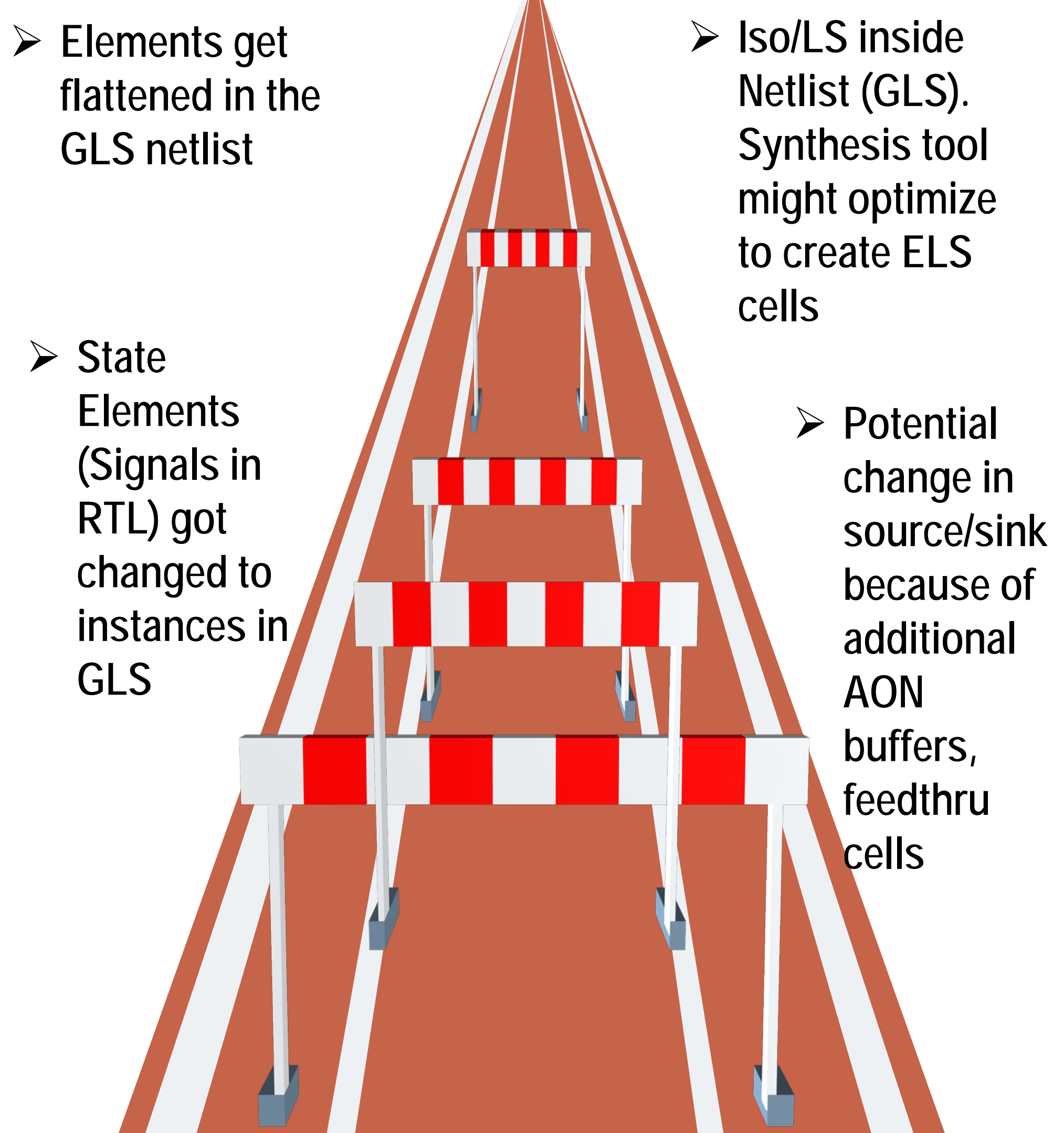
## Benefits of Re-usable UPF



## Difference in Low-Power RTL and GL Netlist



## Challenges in Re-using UPF



## Writing Reusable UPF

### Hier-Path Related Issues in RTL Vs GLS UPF

**Vector signal in UPF** : Writing consistent UPF Definition: reg [2:0] A

Usage:  
 set\_retention -elements {A} >> ❌  
 set\_retention -elements { A[0] A[1] A[2] } >> ✅

**Hier-path scope difference** : Recommendation is to write the elements in the gate-level UPF form

RTL state element	GL State Element	GL UPF	RTL UPF
<pre>module dft(...) reg srpg_flp1; generate begin : srpg always@(posedge clk or negedge rst_t) begin if(!rst_t) srpg_flp1 &lt;= 1'b0; else srpg_flp1 &lt;= enable; end endgenerate</pre>	<pre>module srpg (...) srff_dff srpg_flp1 ( ...); endmodule</pre>	<pre>set_retention ret1 \ -domain pd \ -elements {...dft_inst/srpg/srpg_flp1}</pre>	<pre>set_retention ret1 \ -domain pd \ -elements {...dft_inst/srpg_flp1}</pre>

**Hier-path separator "."** : Separate the generate hierarchies with "." instead of "/"

RTL >> GLS  
 /tb/top/gen[0]/mid\_inst >> /tb/top/./gen[0].mid\_inst

**UPF "find object"**: Use find\_object command wherever possible, since it supports wildcard based search also, so a little change of a name in GLS would not be a problem for RTL UPF.

RTL	GLS	RTL UPF
<pre>for(i=0;i&lt;num;i=i+1) begin:cfg_gen hm_cfg hm_cfg1_mem1(...); hm_cfg hm_cfg1_mem2(...); end</pre>	<pre>hm_cfg \cfg_gen[0].hm_cfg1_ mem1(...); hm_cfg \cfg_gen[0].hm_cfg1_ mem2(...);</pre>	<pre>set ret_exclude_list [ join [find_objects . - transitive true -pattern *hm_cfg1_mem* - object_type instance] ]</pre>

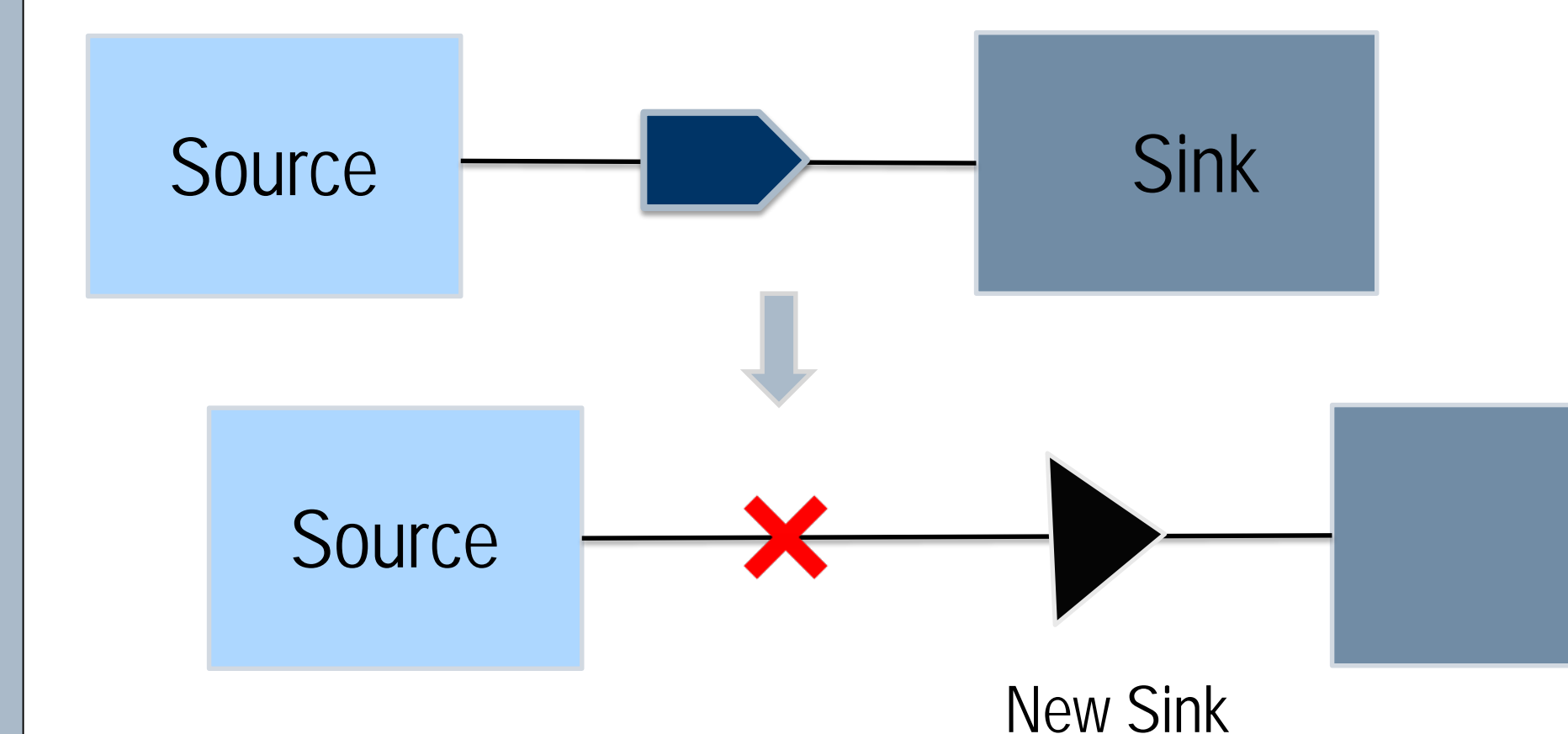
## Writing Reusable UPF contd.

### PA Cell Handling

Iso/LS strategy in RTL UPF is replaced by actual physical cell in the netlist

**Use use\_interface\_cell/map\_\*** : Helps simulation tool to identify and associate the cells correctly with the strategy

**Use most explicit -elements** : Recommendation is to change the RTL UPF (set\_isolation -elements) to be signal wise once it is verified using -source/-sink



Source/sink tends to break in GLS because of additional buffers, AON and other elements in the actual source/sink path

Specify retention elements on signal basis in set\_retention -elements

**Supply connections to PA cells** : Use UPF\_GENERIC

GLS (Iso/LS/ELS are dual-rail, multi-rail or single-rail) Vs RTL (always single rail powered by its strategy) - Mismatch in GLS vs. RTL simulation

UPF GENERICS in conjunction with bind\_checker is very helpful in catching the multi-power rail issues at RTL UPF itself

## Conclusion

