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:

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

<table>
<thead>
<tr>
<th>RTL state element</th>
<th>GL State Element</th>
<th>GL UPF</th>
<th>RTL UPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>module srpg (....)</td>
<td>module srpg_flp1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>generate begin</td>
<td>begin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>```</td>
<td>```</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RAIL TO GLS

- **Diff in Low-Power RTL and GL Netlist**
  - **LANGUAGES**
    - More Abstract
    - Always & Assigns
  - **LANGUAGES**
    - Closer to Silicon
    - Cells & Gates

**Challenges in Re-using UPF**

- Elements get flattened in the GLS netlist
- Iso/LS inside Netlist (GLS). Synthesis tool might optimize to create ELS cells
- State Elements (Signals in RTL) got changed to instances in GLS
- Potential change in source/sink because of additional AON buffers, feedthru cells

**Benefits of Re-usable UPF**

- **Coverage remains intact**
  - UPF is same
  - No need for doing low-power coverage again

- **Logic equivalence – not required**
  - UPF is same
  - No need for equivalence

- **Intact tool specific changes**
  - UPF changes at RTL follows to GLS

- **Concise and easy**
  - UPF not generated by tool