Formal Verification

- Equivalency Checking
  - Are these two designs (that started from the same RTL) equivalent?
  - No new differences will be introduced
  - FormalPro

- Property Checking
  - Will ‘this condition’ ever happen?
  - 0-IN Formal
Where Are Functional Errors Introduced?

- Complex design flows
  - Physical optimization & floorplanning
  - Timing closure
- Product schedules
  - Complete flow iteration impractical
  - Late changes must be made to netlist and layout

Implement change to design netlist
EC performance for Gate Verification when gate simulation is too slow and incomplete

One hour EC versus 1 week simulation time

<table>
<thead>
<tr>
<th>Design Size (K gates)</th>
<th>Gate-Level Simulation 1 CPU</th>
<th>Gate-Level Simulation 4 CPU</th>
<th>FormalPro</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>50</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>300</td>
<td>150</td>
<td>300</td>
<td>150</td>
</tr>
<tr>
<td>400</td>
<td>250</td>
<td>500</td>
<td>250</td>
</tr>
<tr>
<td>500</td>
<td>350</td>
<td>750</td>
<td>350</td>
</tr>
</tbody>
</table>

Graph showing performance comparison between one hour EC and gate-level simulation time.
What Is Equivalence Checking?

- Uses formal techniques to prove that two designs are functionally the same.
- Design A is assumed to be correct.
- Does not replace simulation.

Formal Verification Equivalence Checking

Design A
RTL$_1$ / gates$_1$

Implementation Step

Design B
RTL$_2$ / gates$_2$

A == B?

Yes => ✓
No => ❌

Implementation Step

(square6)

Uses formal techniques to prove that two designs are functionally the same.

Design A is assumed to be correct.

Does not replace simulation.
FormalPro
How it works (compile)

- Compile design data
- Build design correspondence
- Verify the design

Compile
Match
Solve
FormalPro

How it works (match)

- For full verification, all matching must be completed before solving

Compile

Match

Solve

For full verification, all matching must be completed before solving
FormalPro
How it works (solve)

- Matched comparison points become targets
  - Register/Latch outputs
  - Primary outputs
  - Inputs of black boxes
  - Optionally on register inputs

Compile
Match
Solve
Common Issues

- Two Unrelated Designs
- Unmatched Objects
- PowerAware cells
- Large Multipliers (> 30 output bits)
- Merged Operators