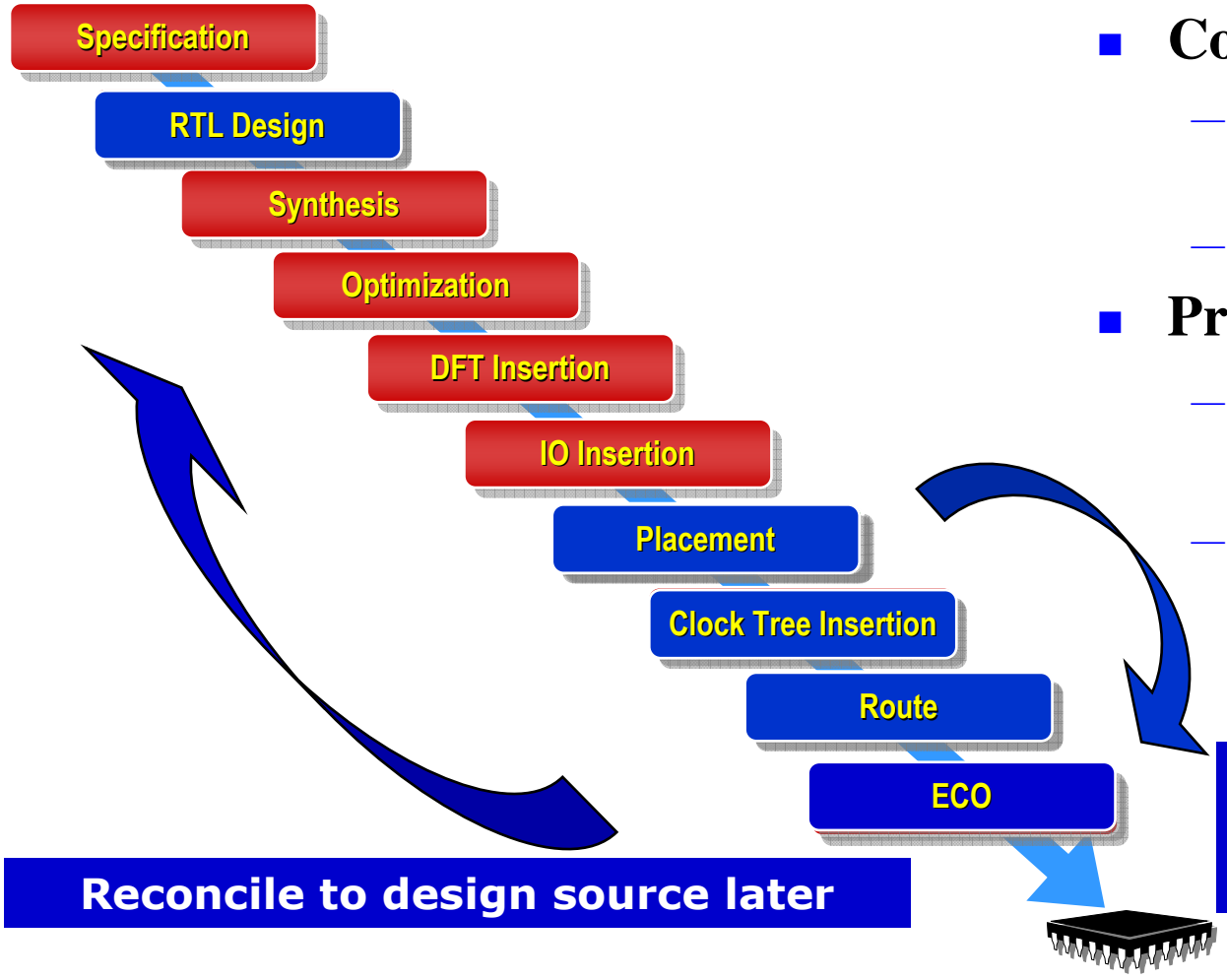
Formal Verification

- **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**

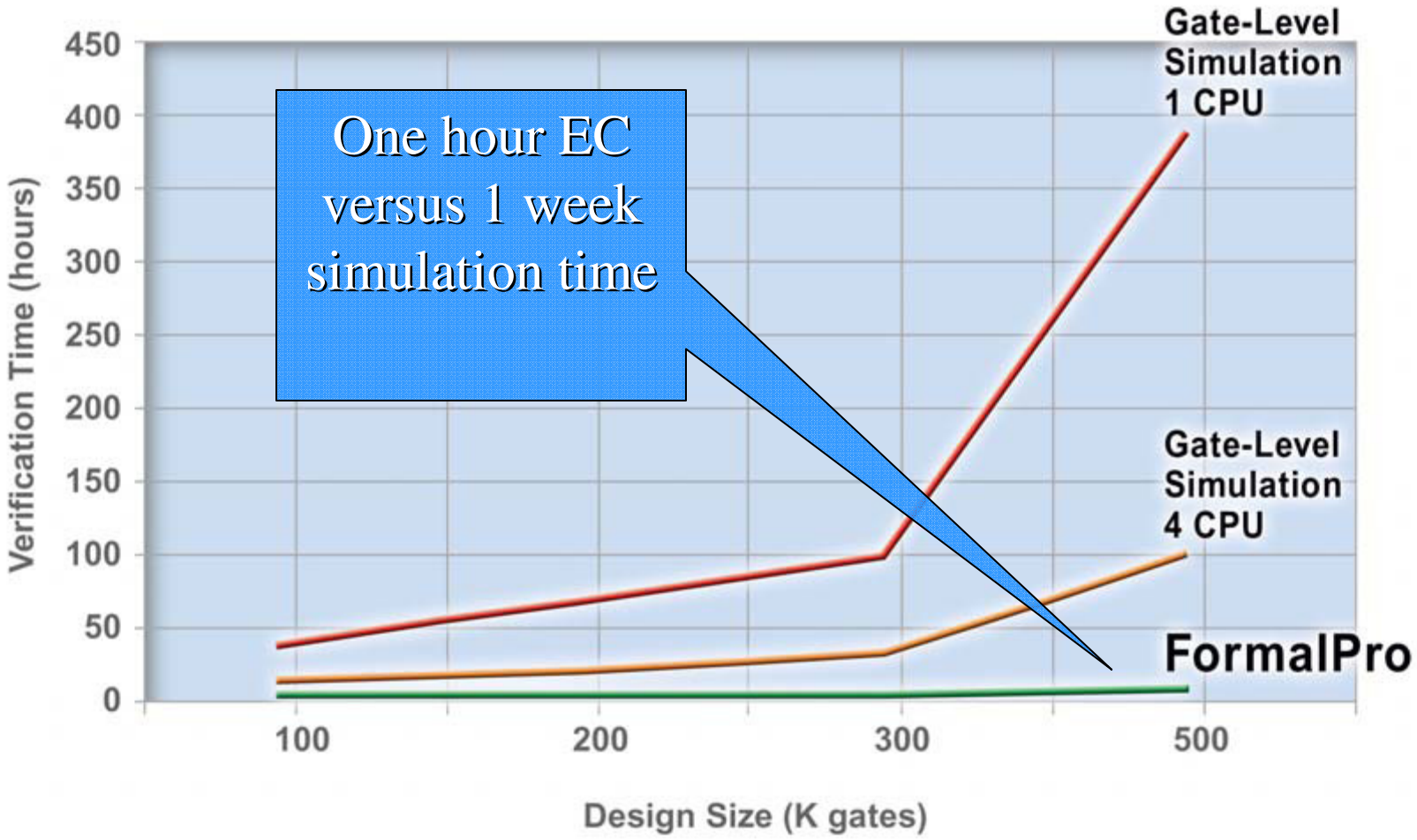
Where Are Functional Errors Introduced?



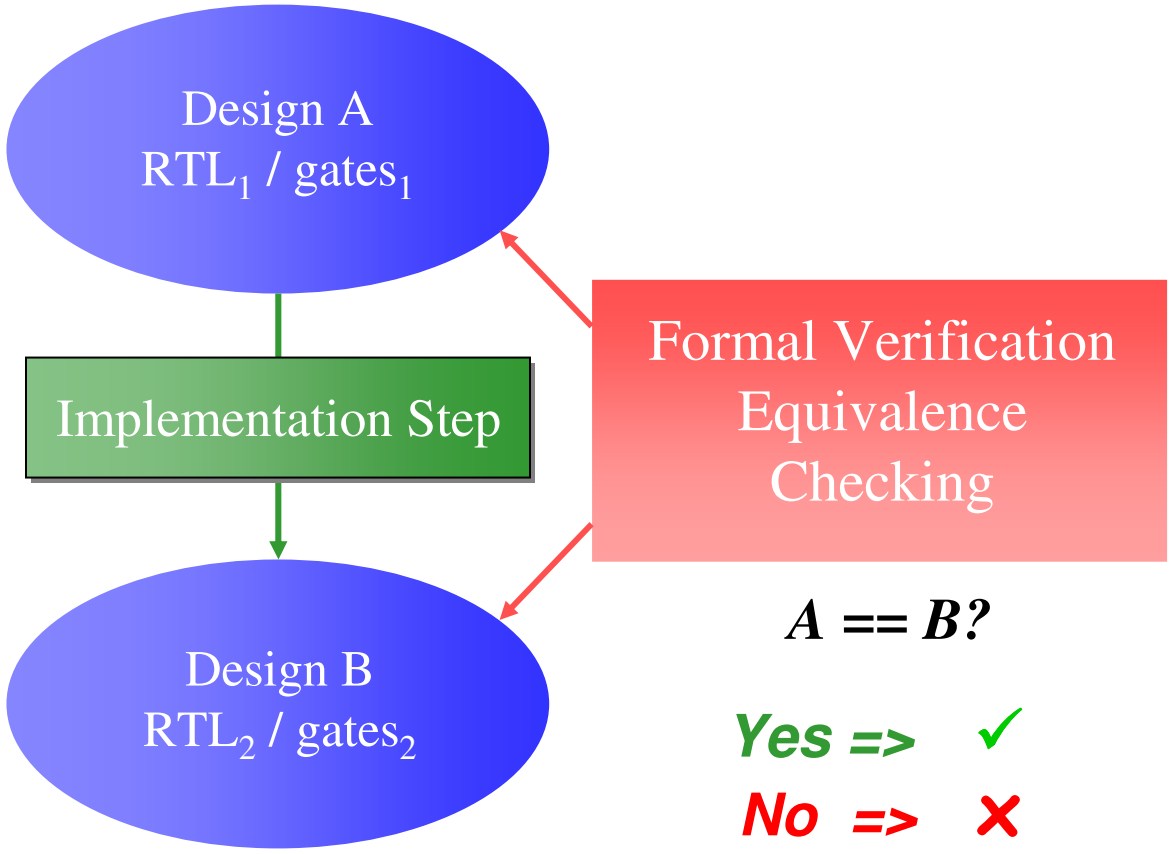
- Complex design flows
  - Physical optimization & floorplanning
  - Timing closure
- Product schedules
  - Complete flow iteration impractical
  - Late changes must be made to netlist and layout

EC performance for Gate Verification

when gate simulation is too slow and incomplete



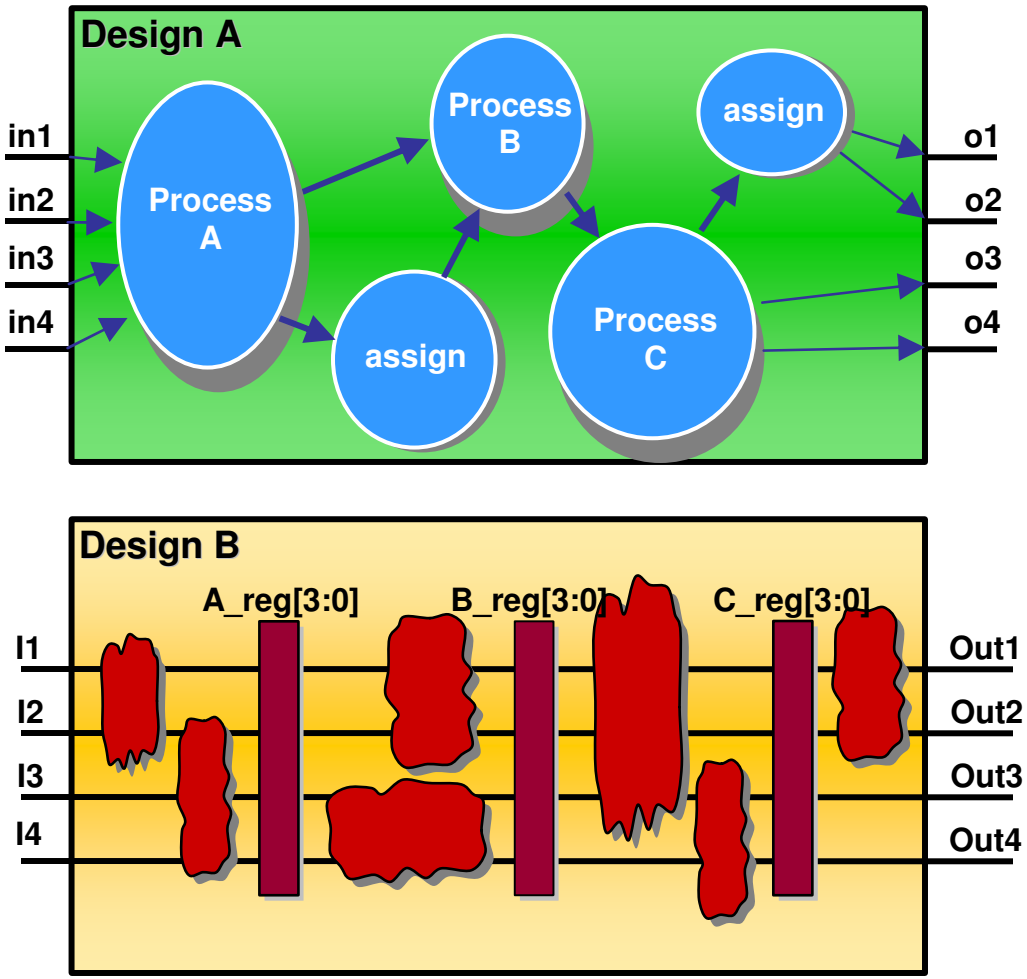
What Is Equivalence Checking?



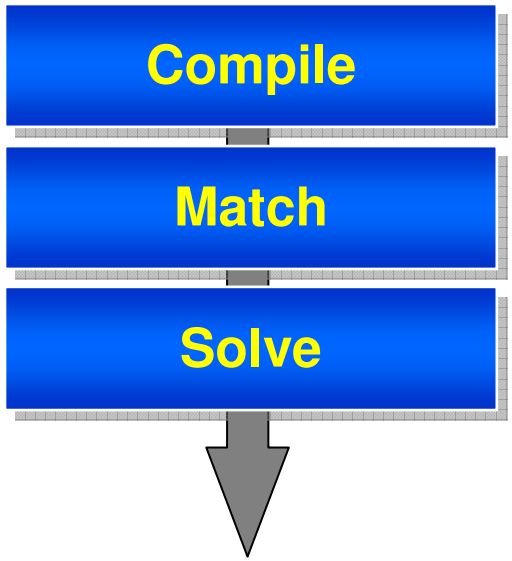
- **Uses formal techniques to <u>prove</u> that two designs are functionally the same**
- **Design A is assumed to be correct**
- **Does not replace simulation**

FormalPro

How it works (compile)

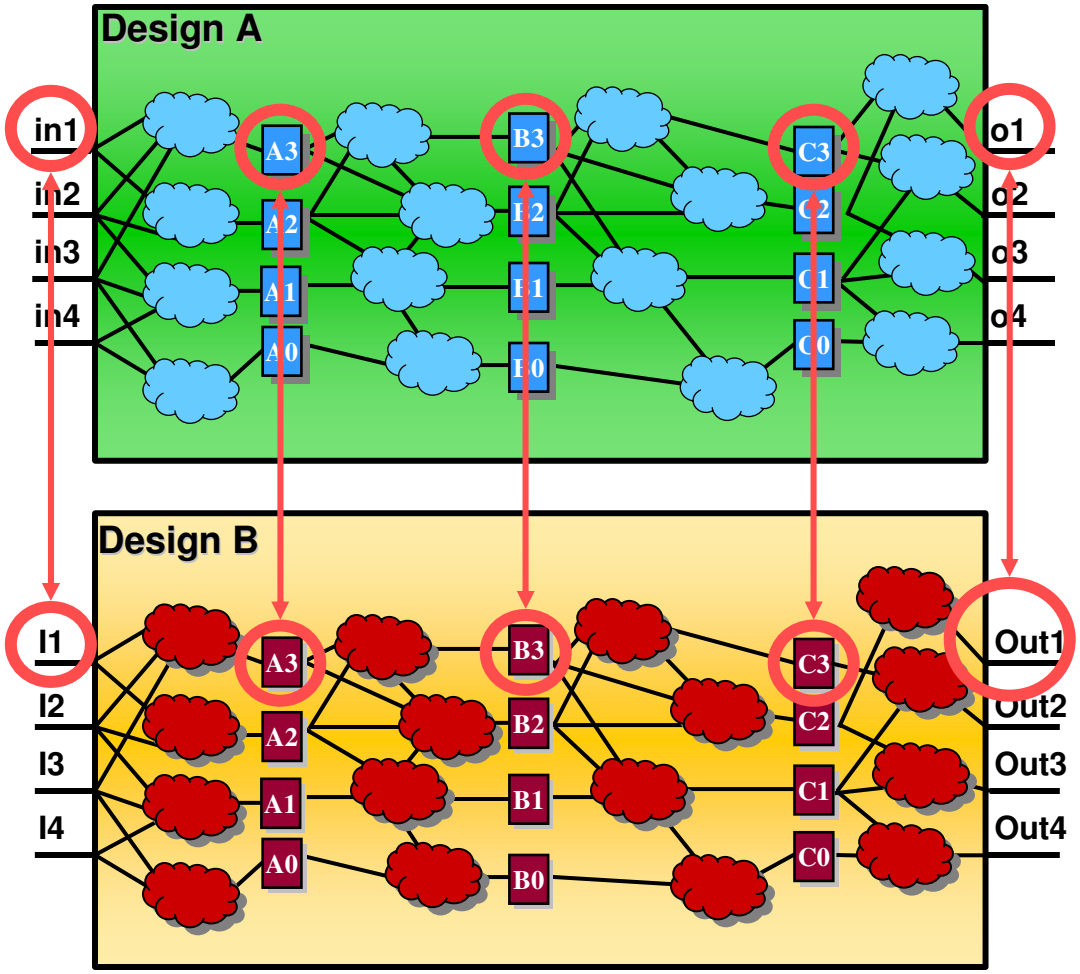


- Compile design data
- Build design correspondence
- Verify the design

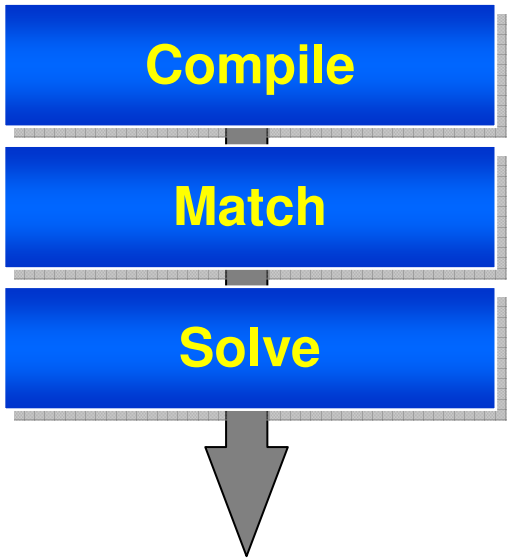


FormalPro

How it works (match)

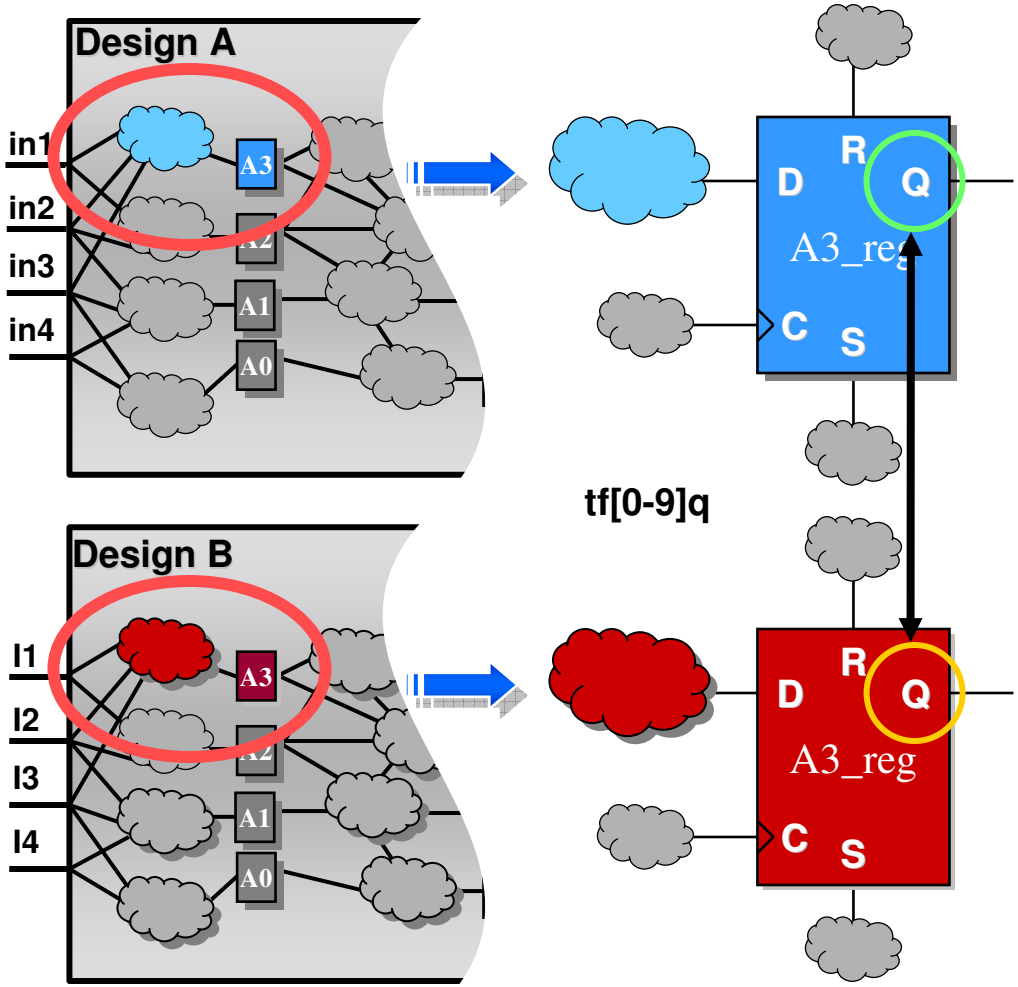


- For full verification, all matching must be completed before solving

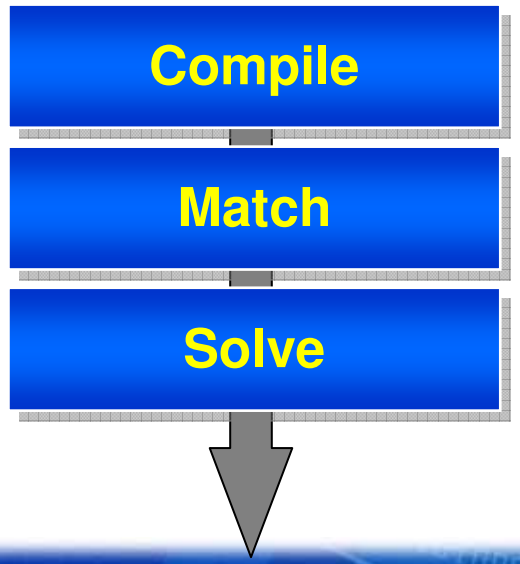


FormalPro

How it works (solve)



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



Common Issues

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